Using the Bay Command Console (AN/BN Routers)

BayRS Version 12.00
BCC Version 3.10

Part No. 117383-A Rev. A
September 1997

Bay Networks

Bay Networks

4401 Great America Parkway
Santa Clara, CA 95054

8 Federal Street
Billerica, MA 01821

Copyright © 1997 Bay Networks, Inc.

All rights reserved. Printed in the USA. September 1997.

The information in this document is subject to change without notice. The statements, configurations, technical data, and recommendations in this document are believed to be accurate and reliable, but are presented without express or implied warranty. Users must take full responsibility for their applications of any products specified in this document. The information in this document is proprietary to Bay Networks, Inc.

The software described in this document is furnished under a license agreement and may only be used in accordance with the terms of that license. A summary of the Software License is included in this document.

Trademarks

AN, BCN, BLN, BN, FRE, and Bay Networks are registered trademarks and ANH, Bay Networks Press, and the Bay Networks logo are trademarks of Bay Networks, Inc.

All other trademarks and registered trademarks are the property of their respective owners.

Restricted Rights Legend

Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Notwithstanding any other license agreement that may pertain to, or accompany the delivery of, this computer software, the rights of the United States Government regarding its use, reproduction, and disclosure are as set forth in the Commercial Computer Software-Restricted Rights clause at FAR 52.227-19.

Statement of Conditions

In the interest of improving internal design, operational function, and/or reliability, Bay Networks, Inc. reserves the right to make changes to the products described in this document without notice.

Bay Networks, Inc. does not assume any liability that may occur due to the use or application of the product(s) or circuit layout(s) described herein.

Portions of the code in this software product are Copyright © 1988, Regents of the University of California. All rights reserved. Redistribution and use in source and binary forms of such portions are permitted, provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that such portions of the software were developed by the University of California, Berkeley. The name of the University may not be used to endorse or promote products derived from such portions of the software without specific prior written permission.

SUCH PORTIONS OF THE SOFTWARE ARE PROVIDED “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

In addition, the program and information contained herein are licensed only pursuant to a license agreement that contains restrictions on use and disclosure (that may incorporate by reference certain limitations and notices imposed by third parties).

Bay Networks, Inc. Software License Agreement

NOTICE: Please carefully read this license agreement before copying or using the accompanying software or installing the hardware unit with pre-enabled software (each of which is referred to as "Software" in this Agreement). BY COPYING OR USING THE SOFTWARE, YOU ACCEPT ALL OF THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT. THE TERMS EXPRESSED IN THIS AGREEMENT ARE THE ONLY TERMS UNDER WHICH BAY NETWORKS WILL PERMIT YOU TO USE THE SOFTWARE. If you do not accept these terms and conditions, return the product, unused and in the original shipping container, within 30 days of purchase to obtain a credit for the full purchase price.

1. License Grant. Bay Networks, Inc. ("Bay Networks") grants the end user of the Software ("Licensee") a personal, nonexclusive, nontransferable license: a) to use the Software either on a single computer or, if applicable, on a single authorized device identified by host ID, for which it was originally acquired; b) to copy the Software solely for backup purposes in support of authorized use of the Software; and c) to use and copy the associated user manual solely in support of authorized use of the Software by Licensee. This license applies to the Software only and does not extend to Bay Networks Agent software or other Bay Networks software products. Bay Networks Agent software or other Bay Networks software products are licensed for use under the terms of the applicable Bay Networks, Inc. Software License Agreement that accompanies such software and upon payment by the end user of the applicable license fees for such software.

2. Restrictions on use; reservation of rights. The Software and user manuals are protected under copyright laws. Bay Networks and/or its licensors retain all title and ownership in both the Software and user manuals, including any revisions made by Bay Networks or its licensors. The copyright notice must be reproduced and included with any copy of any portion of the Software or user manuals. Licensee may not modify, translate, decompile, disassemble, use for any competitive analysis, reverse engineer, distribute, or create derivative works from the Software or user manuals or any copy, in whole or in part. Except as expressly provided in this Agreement, Licensee may not copy or transfer the Software or user manuals, in whole or in part. The Software and user manuals embody Bay Networks' and its licensors' confidential and proprietary intellectual property. Licensee shall not sublicense, assign, or otherwise disclose to any third party the Software, or any information about the operation, design, performance, or implementation of the Software and user manuals that is confidential to Bay Networks and its licensors; however, Licensee may grant permission to its consultants, subcontractors, and agents to use the Software at Licensee's facility, provided they have agreed to use the Software only in accordance with the terms of this license.

3. Limited warranty. Bay Networks warrants each item of Software, as delivered by Bay Networks and properly installed and operated on Bay Networks hardware or other equipment it is originally licensed for, to function substantially as described in its accompanying user manual during its warranty period, which begins on the date Software is first shipped to Licensee. If any item of Software fails to so function during its warranty period, as the sole remedy Bay Networks will at its discretion provide a suitable fix, patch, or workaround for the problem that may be included in a future Software release. Bay Networks further warrants to Licensee that the media on which the Software is provided will be free from defects in materials and workmanship under normal use for a period of 90 days from the date Software is first shipped to Licensee. Bay Networks will replace defective media at no charge if it is returned to Bay Networks during the warranty period along with proof of the date of shipment. This warranty does not apply if the media has been damaged as a result of accident, misuse, or abuse. The Licensee assumes all responsibility for selection of the Software to achieve Licensee's intended results and for the installation, use, and results obtained from the Software. Bay Networks does not warrant a) that the functions contained in the software will meet the Licensee's requirements, b) that the Software will operate in the hardware or software combinations that the Licensee may select, c) that the operation of the Software will be uninterrupted or error free, or d) that all defects in the operation of the Software will be corrected. Bay Networks is not obligated to remedy any Software defect that cannot be reproduced with the latest Software release. These warranties do not apply to the Software if it has been (i) altered, except by Bay Networks or in accordance with its instructions; (ii) used in conjunction with another vendor's product, resulting in the defect; or (iii) damaged by improper environment, abuse, misuse, accident, or negligence. THE FOREGOING WARRANTIES AND LIMITATIONS ARE EXCLUSIVE REMEDIES AND ARE IN LIEU OF ALL OTHER WARRANTIES EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Licensee is responsible for the security of its own data and information and for maintaining adequate procedures apart from the Software to reconstruct lost or altered files, data, or programs.

4. Limitation of liability. IN NO EVENT WILL BAY NETWORKS OR ITS LICENSORS BE LIABLE FOR ANY COST OF SUBSTITUTE PROCUREMENT; SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES; OR ANY DAMAGES RESULTING FROM INACCURATE OR LOST DATA OR LOSS OF USE OR PROFITS ARISING OUT OF OR IN CONNECTION WITH THE PERFORMANCE OF THE SOFTWARE, EVEN IF BAY NETWORKS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL THE LIABILITY OF BAY NETWORKS RELATING TO THE SOFTWARE OR THIS AGREEMENT EXCEED THE PRICE PAID TO BAY NETWORKS FOR THE SOFTWARE LICENSE.

5. Government Licensees. This provision applies to all Software and documentation acquired directly or indirectly by or on behalf of the United States Government. The Software and documentation are commercial products, licensed on the open market at market prices, and were developed entirely at private expense and without the use of any U.S. Government funds. The license to the U.S. Government is granted only with restricted rights, and use, duplication, or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c)(1) of the Commercial Computer Software—Restricted Rights clause of FAR 52.227-19 and the limitations set out in this license for civilian agencies, and subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause of DFARS 252.227-7013, for agencies of the Department of Defense or their successors, whichever is applicable.

6. Use of Software in the European Community. This provision applies to all Software acquired for use within the European Community. If Licensee uses the Software within a country in the European Community, the Software Directive enacted by the Council of European Communities Directive dated 14 May, 1991, will apply to the examination of the Software to facilitate interoperability. Licensee agrees to notify Bay Networks of any such intended examination of the Software and may procure support and assistance from Bay Networks.

7. Term and termination. This license is effective until terminated; however, all of the restrictions with respect to Bay Networks' copyright in the Software and user manuals will cease being effective at the date of expiration of the Bay Networks copyright; those restrictions relating to use and disclosure of Bay Networks' confidential information shall continue in effect. Licensee may terminate this license at any time. The license will automatically terminate if Licensee fails to comply with any of the terms and conditions of the license. Upon termination for any reason, Licensee will immediately destroy or return to Bay Networks the Software, user manuals, and all copies. Bay Networks is not liable to Licensee for damages in any form solely by reason of the termination of this license.

8. Export and Re-export. Licensee agrees not to export, directly or indirectly, the Software or related technical data or information without first obtaining any required export licenses or other governmental approvals. Without limiting the foregoing, Licensee, on behalf of itself and its subsidiaries and affiliates, agrees that it will not, without first obtaining all export licenses and approvals required by the U.S. Government: (i) export, re-export, transfer, or divert any such Software or technical data, or any direct product thereof, to any country to which such exports or re-exports are restricted or embargoed under United States export control laws and regulations, or to any national or resident of such restricted or embargoed countries; or (ii) provide the Software or related technical data or information to any military end user or for any military end use, including the design, development, or production of any chemical, nuclear, or biological weapons.

9. General. If any provision of this Agreement is held to be invalid or unenforceable by a court of competent jurisdiction, the remainder of the provisions of this Agreement shall remain in full force and effect. This Agreement will be governed by the laws of the state of California.

Should you have any questions concerning this Agreement, contact Bay Networks, Inc., 4401 Great America Parkway, P.O. Box 58185, Santa Clara, California 95054-8185.

LICENSEE ACKNOWLEDGES THAT LICENSEE HAS READ THIS AGREEMENT, UNDERSTANDS IT, AND AGREES TO BE BOUND BY ITS TERMS AND CONDITIONS. LICENSEE FURTHER AGREES THAT THIS AGREEMENT IS THE ENTIRE AND EXCLUSIVE AGREEMENT BETWEEN BAY NETWORKS AND LICENSEE, WHICH SUPERSEDES ALL PRIOR ORAL AND WRITTEN AGREEMENTS AND COMMUNICATIONS BETWEEN THE PARTIES PERTAINING TO THE SUBJECT MATTER OF THIS AGREEMENT. NO DIFFERENT OR ADDITIONAL TERMS WILL BE ENFORCEABLE AGAINST BAY NETWORKS UNLESS BAY NETWORKS GIVES ITS EXPRESS WRITTEN CONSENT, INCLUDING AN EXPRESS WAIVER OF THE TERMS OF THIS AGREEMENT.

Contents

About This Guide

Conventions xiv
Acronyms xv
Ordering Bay Networks Publications xvi
Bay Networks Customer Service xvi
How to Get Help xvii

Chapter 1
Overview of the BCC Interface

Platform Requirements 1-2
Configurable Objects 1-2
Number of BCC Sessions 1-2
Terminology and Concepts 1-3
Using Abbreviations and Acronyms 1-7

Chapter 2
Learning to Use the BCC Interface

Entering and Exiting the BCC Interface 2-1
Using the BCC Configuration Hierarchy 2-3
Locating Configured Objects 2-5
Displaying Context 2-6
Context-Sensitive Prompts 2-6

Navigating in Configuration Mode 2-7
Navigating with the back Command 2-7
Navigating with Configuration Commands 2-8
Moving Back One or More Levels 2-8
Moving Back to Root (box) Level 2-8
Moving Forward from Root (box) Level 2-9
Specifying a Full Path 2-10
Specifying a Shortened Path 2-10
Displaying Online Help 2-12
Using the BCC Help File 2-25
Displaying Configuration Data 2-26
Displaying the Total Device Configuration 2-26
Displaying the Configuration of One Object 2-28
Displaying Binary Configuration Files as BCC Syntax 2-29

Chapter 3
Entering Commands

Command Input Features 3-2
Using Command Abbreviations 3-2
Recalling Commands 3-2
Entering Commands from a File 3-3
Entering Multiple Commands per Line 3-4
Continuing a Command Line 3-4
Entering Comments 3-6
System Commands 3-6
Configuration Commands 3-7
Using Basic (Full) Syntax 3-7
Using Default Syntax 3-8
Using Abbreviated Syntax 3-9
Specifying Parameter Values 3-10
Required, Derived, and Other Parameters 3-10
Specifying Multiple Parameter-Value Pairs 3-11
Command Operators 3-11
Editing Command Lines 3-13
Saving Commands on a Workstation 3-14
Saving the Active Configuration to an NVFS Volume 3-14

Chapter 4
Configuring a Network Device

Creating and Modifying a Device Configuration 4-1
Entering Configuration Commands from a File 4-16
Disabling a Configured Object 4-17
Enabling a Configured Object 4-18
Deleting a Configured Object 4-19
Configuration Command Responses 4-20

Chapter 5
Configuration Examples

Identifying Installed Link Modules 5-2
Configuring an Ethernet Interface with IP, ARP, and RIP 5-4
Configuring a HSSI Port with IP 5-7
Configuring a Token Ring Interface with IP and RIP 5-10
Configuring PPP and IP on a Synchronous Interface 5-12
Configuring a FDDI Interface with IP and RIP 5-13
Configuring OSPF and BGP 5-15
Configuring Telnet, FTP, SNMP, and NTP 5-23

Appendix A
Configuration Tree

Appendix B
System Commands

Appendix C
TCL Support

Index

Figures

Figure 1-1. The Technician Interface and the BCC Interface 1-1
Figure 1-2. Configuring IP and RIP on an Ethernet Interface 1-5
Figure 2-1. Sample BCC Configuration (AN/BN Router) 2-4
Figure 2-2. Object Location or Context 2-5
Figure 2-3. Moving Away from Root (box) Level 2-9
Figure 2-4. Specifying a Full Path 2-10
Figure 2-5. Navigating with the BCC Backward (Recursive) Search Feature 2-11
Figure 2-6. Objects to Configure at the Next (Branch) Level 2-23
Figure 4-1. Sample BCC Configuration (BCN Router) 4-3
Figure 4-2. Typical BCC Configuration Cycle 4-4

Tables

Table 3-1. BCC Command Operators (Methods) 3-12
Table 3-2. Keystrokes for Editing BCC Command Lines 3-13
Table B-1. AN/BN Router System Commands B-1

About This Guide

If you are responsible for configuring and managing Bay Networks® AN® and BN® (BCN® and BLN®) routers, read this guide to learn how to use the Bay Command Console (BCC™).

If you want to	Go to
Learn about BCC concepts and terminology	Chapter 1
Learn how to perform basic BCC operations	Chapter 2
Learn how to enter BCC commands	Chapter 3
Learn how to use the BCC to configure an AN or BN router	Chapter 4
Study examples that illustrate how to configure, navigate, get help, and perform other system tasks on an AN or BN router	Chapter 5

To use the BCC, you should have some experience supporting a multivendor internetworking system. You should be able to perform network device configuration, maintenance, and troubleshooting.

For additional sources of information about configuring AN and BN routers, see the task map for this release.

Note: Because the BCC makes real-time changes to a device configuration, Bay Networks recommends that you first learn about BCC behavior on a device not connected to your production network. After you become comfortable with using the BCC, run it on a device in your production network.

Conventions

angle brackets (< >)	Indicate a variable in a command line. The word or words inside the angle brackets describe the type of variable (for example, <*host-address*>). Do not type the angle brackets when entering an actual value for a variable. Example: if command syntax is **ping** *<ip-address>*, enter **ping 192.32.10.12**
`bold text`	Indicates text (usually commands) that you enter at the BCC command line prompt. Example: Enter **`reset`**.

braces ({ })	Enclose a list of
	
	• Command keywords or configuration parameter values, one of which you can enter
	
	• BCC instance IDs of configured objects
	
	Example:
	`{arp/1.2.3.4/1 rip/1.2.3.4}`
	
	• A literal string, such as the literal value of the console prompt
	
	Example:
	`prompt {[%slot%:1]$}`
	
	(Yields a prompt such as "`[9:1]$`".)

brackets ([ ])	Indicate command keywords, arguments, or filters not required (taken as optional command input) by the BCC.
ellipsis points (. . .)	Repeat the last element of the command as needed. Example: **ethernet/2/1** [<*parameter*> <*value*>] **. . .** Translating this syntax to a specific command: **ethernet/2/1 bofl-retries 2 hardware-filter enabled**

italic text	Indicates variable values in command syntax descriptions, new terms, file and directory names, and book titles.
quotation marks (“ ”)	Indicate a literal string in a command line, the title of a chapter, or a section within a book.
`screen text`	Indicates BCC or Technician Interface output to a console or Telnet screen, for example, prompts, system messages, statistical data, and configuration data.
vertical line (|)	Separates choices for required or optional command keywords and arguments. You must enter only one of the choices available. Do not type the vertical line when entering a command. Example: If the command syntax is **show ip routes** | **rip**, you enter either **show ip routes** or **show ip rip**, but not both.

Acronyms

ARP	Address Resolution Protocol
BN	Backbone Node
IP	Internet Protocol
LAN	local area network
MAC	media access control
OSPF	Open Shortest Path First
RIP	Routing Information Protocol
SNMP	Simple Network Management Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TFTP	Trivial File Transfer Protocol
WAN	wide area network

Ordering Bay Networks Publications

To purchase additional copies of this document or other Bay Networks publications, order by part number from Bay Networks Press™ at the following numbers:

- Phone--U.S./Canada: 888-422-9773
- Phone--International: 510-490-4752
- FAX--U.S./Canada and International: 510-498-2609

The Bay Networks Press catalog is available on the World Wide Web at *support.baynetworks.com/Library/GenMisc*. Bay Networks publications are available on the World Wide Web at *support.baynetworks.com/Library/tpubs*.

Bay Networks Customer Service

You can purchase a support contract from your Bay Networks distributor or authorized reseller, or directly from Bay Networks Services. For information about, or to purchase a Bay Networks service contract, either call your local Bay Networks field sales office or one of the following numbers:

Region	Telephone number	Fax number
United States and Canada	800-2LANWAN; then enter Express Routing Code (ERC) 290, when prompted, to purchase or renew a service contract 978-916-8880 (direct)	978-916-3514
Europe	33-4-92-96-69-66	33-4-92-96-69-96
Asia/Pacific	61-2-9927-8888	61-2-9927-8899
Latin America	561-988-7661	561-988-7550

Information about customer service is also available on the World Wide Web at *support.baynetworks.com*.

How to Get Help

If you purchased a service contract for your Bay Networks product from a distributor or authorized reseller, contact the technical support staff for that distributor or reseller for assistance.

If you purchased a Bay Networks service program, call one of the following Bay Networks Technical Solutions Centers:

Technical Solutions Center	Telephone number	Fax number
Billerica, MA	800-2LANWAN	978-916-3514
Santa Clara, CA	800-2LANWAN	408-495-1188
Valbonne, France	33-4-92-96-69-68	33-4-92-96-69-98
Sydney, Australia	61-2-9927-8800	61-2-9927-8811
Tokyo, Japan	81-3-5402-0180	81-3-5402-0173

Chapter 1
Overview of the BCC Interface

The BCC is a command-line interface for configuring Bay Networks devices. After logging on to a device, you access the BCC by entering the **bcc** command at the Technician Interface prompt (Figure 1-1).

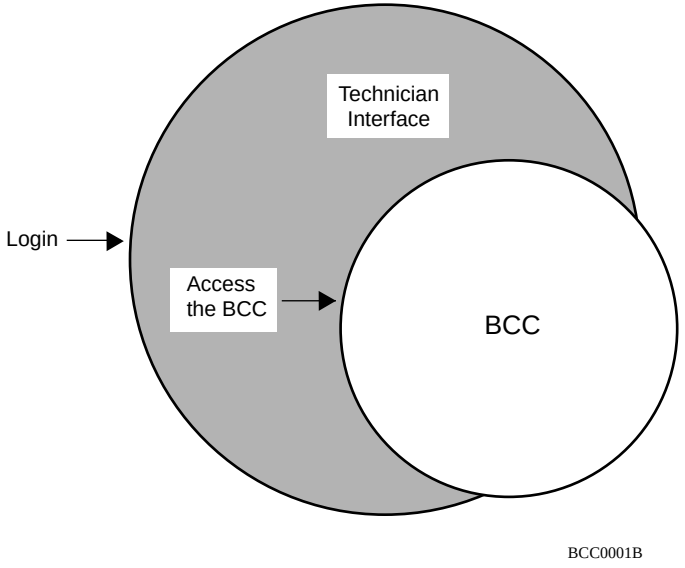


Figure 1-1. The Technician Interface and the BCC Interface

From the bcc> prompt, you can

- Run any system command not classified as "Technician Interface only." (See Appendix B, "System Commands.")
- Use configuration mode to perform tasks such as creating or deleting IP interfaces on the router. Enter BCC configuration mode by entering the **config** command at the bcc> prompt.

For tasks related to device management (managing files on the router, viewing statistics or the event log, and so on), you can enter Technician Interface commands directly at the BCC command-line prompt. (For a complete list of system commands, refer to Appendix B.)

Note: For more information about Technician Interface commands, refer to *Using Technician Interface Software.*

Platform Requirements

The BCC software requires 8 MB of dynamic RAM (DRAM) to run on AN and BN (BLN/BCN) router platforms. In addition, a minimum of 1.5 MB of DRAM must be available when you start the BCC interface.

Configurable Objects

Refer to the Release Notes for Router Software 12.00 for the most up-to-date information about what services you can configure using the BCC on AN and BN routers. (You can also obtain this information by entering the **help tree** command at any BCC prompt.)

Number of BCC Sessions

You can open one BCC session per slot. For the AN router (a single-slot device), this means you can open only one BCC session at a time. For a BCN or BLN router (a multiple-slot device), you can open multiple concurrent BCC sessions, but with only one session active per slot.

Terminology and Concepts

The following list of terms and concepts defines key functional and user-oriented aspects of the BCC interface:

Object -- A data structure representing a configurable physical or logical entity such as a physical interface or a protocol on an interface. Every configurable object belongs to a specific *class* that defines its characteristics.

Class -- A template for a configurable object (such as Ethernet or the protocol IP). When you add a new object to the configuration of a device, the BCC creates a copy (an *instance*) of the appropriate template. Each instance is an object uniquely identifiable within the total device configuration.

Parameters -- Attributes or properties of a configurable object. For example, some parameters of Ethernet are

- slot and connector (describing the location of the Ethernet interface)
- bofl-retries (describing one behavioral aspect of the Ethernet interface)

Required parameters -- A minimum set of parameters for which the BCC requires you to supply values. For example, the required parameters of a physical port are **slot** and **connector**.

Derived parameters -- Parameters for which the BCC supplies a value. For example, a derived parameter of OSPF (global) is **router-id**. In this case, the BCC derives a value for router-id from the address of the first IP interface configured on the device.

Optional parameters -- The set of parameters for which you can optionally specify customized values, replacing any default values set by the system. For example, an optional parameter of an Ethernet interface is **bofl-retries**. This parameter normally has a system default value of **5** (5 retries), but you can change this to another numeric value.

BCC instance identifier -- Uniquely identifies a single instance of an object configured on a Bay Networks device. The BCC instance ID consists typically of the name of an object, combined with the values you specify for its required parameters. For example, the BCC instance ID for an Ethernet interface consists of **ethernet/***<slot>***/***<connector>*, as in **ethernet/2/1**.

For some objects, the BCC automatically appends an internally generated number to ensure uniqueness of the BCC instance ID. For example, the BCC creates an instance ID for ARP on IP based on the object name (arp), plus the address of the underlying IP interface (1.2.3.4), plus an internally generated integer, resulting in an ID such as **arp/1.2.3.4/1**.

In other rare cases, a configurable object may also have required parameters that do not become part of its BCC instance ID. For example, the global OSPF object has a required router-id parameter that does not become part of the instance ID. (Many such required parameters end in "-id," such as *global-id* and *router-id*.)

Each object has its own requirements for unique instance identification within the total device configuration.

Configuration hierarchy -- Classes (templates for creating objects) exist within a tree hierarchy. Just as a file system has a root directory, subdirectores, and files, the BCC configuration system has a root-level object (called "box") and branch objects (such as interfaces and protocols) that fan out from the root level in a tree hierarchy.

The BCC configuration command hierarchy varies according to the type of network device (for example, a particular type of router, hub, or switch). However, BCC commands, such as **?**, **help tree**, and **back**, enable you to efficiently discover and navigate the configuration hierarchy. Appendix A shows the configuration hierarchy for Bay Networks AN and BN routers. You can also view the BCC configuration hierarchy by entering the **help tree** command at any BCC prompt.

You create a customized configuration by using BCC commands to create new objects, and to modify or delete objects in an existing configuration. You begin at *box* level in BCC configuration mode and navigate to (or create new) objects in the device configuration tree.

For example, you can use BCC commands to add a new physical interface (such as Ethernet) on *box*, add IP to the Ethernet interface, and then add RIP to IP on that interface. Figure 1-2 shows the sequence of commands necessary to build this configuration.

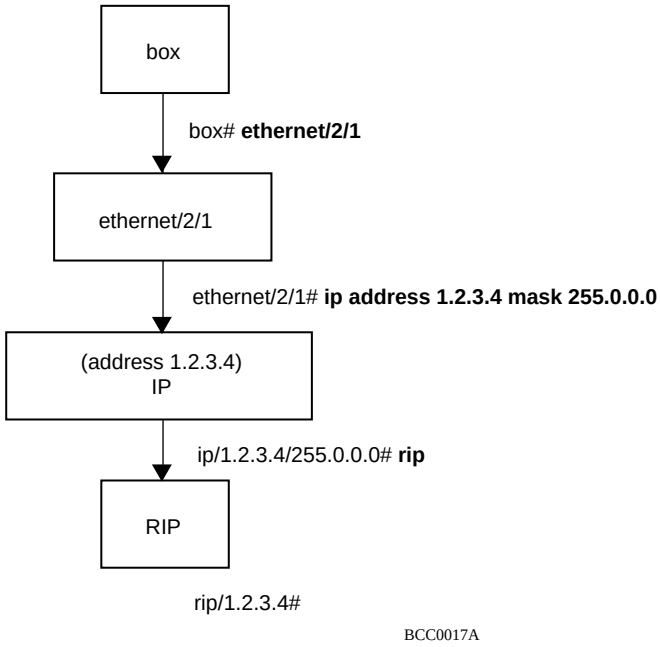


Figure 1-2. Configuring IP and RIP on an Ethernet Interface

Context -- Your working *location* or *level* within the BCC configuration tree. Just as a UNIX file system has a current working directory within which you can add, modify, or delete files, the BCC configuration tree has a current working *context*, within which you can add, modify, disable, reenable, or delete objects.

The BCC indicates *context* by displaying a context-sensitive prompt in configuration mode. The prompt identifies the configured object at your current working location or level within the configuration hierarchy. For example, after logging in as Manager, then configuring or navigating to the Ethernet interface on slot 2 connector 1, the BCC displays the following prompt:

```
ethernet/2/1#
```

Has -- A parameter with a value determined by the branch objects configured one level lower on the device configuration tree. For example, in Figure 1-2, "box" *has* a branch object known uniquely to the BCC as ethernet/2/1. The configured object ethernet/2/1, in turn, *has* a branch object known uniquely as ip/1.2.3.4/255.0.0.0, which in turn has a *leaf object* called rip/1.2.3.4. (*Leaf objects* are the last objects configured on any branch of the device configuration tree.)

Box -- The BCC uses the name *box* to identify the chassis for a Bay Networks nonstackable device. Every box has a type parameter; the value assigned to the type parameter identifies the type of Bay Networks device. For example, the type parameter has the value “an” for an AN router, and “frecn” for a BCN router with a FRE controller module.

Global/Box-wide objects -- Objects that provide services uniformly to all slots of a network device (box-wide); for example, global IP, BGP, TCP, SNMP, FTP, TFTP, and Telnet. Some protocols, such as IP, RIP, and OSPF, have global and interface-level objects.

Board -- The BCC uses the name *board* to identify any logic or circuit board in a Bay Networks device. Each board (or module) typically occupies a *slot* in a network device. On some Bay Networks products, one board may contain another board such as an RMON probe or a data collection module (DCM). All board objects have a *type* parameter that identifies the module type. For example, “qenf” is the value of the type parameter for a Quad Ethernet with Filters module.

Note: Refer to the Release Notes for module descriptions based on the literal value of the *type* parameter for any board object. For example, using the table from the Release Notes, a board type of “qenf” translates to a Quad (four-port) Ethernet with hardware filters module.

Slot -- A location, as well as a physical and electrical means, for attaching modules to logic and power connections available on the device chassis.

Connector -- The physical and electrical means to interconnect an interface module in a network device directly or indirectly to a physical-layer network medium.

Line -- A physical (and in some cases, logical) circuit identified typically by means of a slot, connector, interface type (CSMACD, SYNC, FDDI, and so on), and, where applicable, a channel number (such as with TI/E1 interface types).

Port -- An interface object defined by its type (for example, an Ethernet port) and location (slot and connector) within a network device. On a network device or a user end station, a port is also a logical point of termination for data sent or received by a specific protocol or application. For example, a UNIX workstation receives syslog messages from a remote device at UDP logical port number 162.

Interface, physical -- Circuitry and digital logic associated with the interconnection between a physical network medium (such as Ethernet) and a higher-layer protocol entity (such as IP).

Interface, logical -- An addressable entity for originating and terminating connections across an IP network.

Protocol -- A configurable object that typically provides data link, network, transport, session, application, or management layer services on a network device. A protocol on a router may provide services box-wide (across all slots, such as global IP) or on a per-interface basis (such as interface IP) on a router platform.

System commands -- Enable you to perform system administration tasks from any command-line prompt. For a complete list of system commands available at any level of the BCC interface, enter **?** at any BCC command-line prompt.

Using Abbreviations and Acronyms

Words that represent objects, parameters, and certain parameter values for command input or output are

- Industry-accepted words or standard abbreviations and acronyms
- Standard Bay Networks abbreviations and acronyms

For command input, the BCC interface allows you to shorten existing object and parameter names, for example, **e** or **eth** for **ethernet**. This is the “minimum to distinguish” feature of the BCC interface.

Example

Three objects that begin with “t” (**tftp**, **telnet**, and **tokenring**) exist at the root (box) level of the AN/BN router configuration tree. So that the BCC knows which of these objects you want to configure, you must minimally enter at least two letters (for example, **te** for **telnet**) at the box# prompt.

For standard (console or Telnet screen) output, the BCC allows somewhat greater flexibility in the use of abbreviations and acronyms, and allows the use of uppercase characters.

Chapter 2
Learning to Use the BCC Interface

This chapter provides primary instructions on how to use the BCC interface to complete any of the following tasks:

Topic	Page
Entering and Exiting the BCC Interface	2-1
Using the BCC Configuration Hierarchy	2-3
Navigating in Configuration Mode	2-7
Displaying Online Help	2-12
Using the BCC Help File	2-25
Displaying Configuration Data	2-26

Entering and Exiting the BCC Interface

To access the BCC interface on an AN or BN (BCN/BLN) router, first open a Technician Interface session with the target device from

- An ASCII terminal (for example, a VT-100 device) locally attached to the router console port
- A workstation or PC running terminal emulation software and locally attached to the router console port
- A remote workstation or PC running Telnet

To enter and exit the BCC:

1. **Access the Technician Interface on the router by entering the Manager or User command at the `Login` prompt that appears on your Telnet or console display.**

   After you start the BCC interface, the Manager login allows you to enter any system command. A Manager login also allows, by default, read-write access to the device configuration. The User login allows you to enter only user-level system commands, and allows read-only access to the device configuration. With read-only access, you can display or navigate the existing configuration, but you cannot modify the configuration.

2. **Start the BCC interface by entering bcc at the Technician Interface prompt.**

```
Router1> bcc

Welcome to Bay Command Console!

     * To enter configuration mode, type config

     * To list all system commands, type ?

     * To exit the BCC, type exit
bcc>
```

3. **Start BCC configuration mode by entering config at the `bcc>` prompt.**

```
bcc>  config
box#
```

   You enter configuration mode at the root (box) level of the BCC configuration tree. The prompt ends with a pound (#) symbol if you have read-write privileges (Manager only), or with a greater than symbol (>) if you have read-only privileges (Manager or User).

   If you enter BCC configuration mode as Manager and want to change your privilege level for the current session from read-write to read-only, enter **config -read-only**. To change Manager privileges back to read-write, enter **config -read-write**. You cannot change your privilege level from read-only to read-write if you logged in as User.

Caution: When you enter BCC configuration commands with read-write (Manager) access privileges, you immediately modify the device configuration. This is equivalent to dynamic mode configuration with the Bay Networks Site Manager graphical user interface.

4. **When you finish using BCC configuration mode, enter the exit command at any prompt.**

```
box#  exit
bcc>
```

5. **When you finish using the BCC, enter the exit command at the** `bcc>` **prompt.**

```
bcc>  exit
Router1>
```

Exiting the BCC returns you to the Technician Interface prompt.

If you need more detailed information about Technician Interface access, login, or logout procedures, refer to *Using Technician Interface Software*.

Using the BCC Configuration Hierarchy

The BCC interface supports a set of configuration commands based on the contents of the configuration tree for each device. The contents of the tree is different for different device types. (See “Terminology and Concepts” in Chapter 1 for more information about the BCC configuration hierarchy.)

Figure 2-1 illustrates a sample BCC configuration for an AN or BN router.

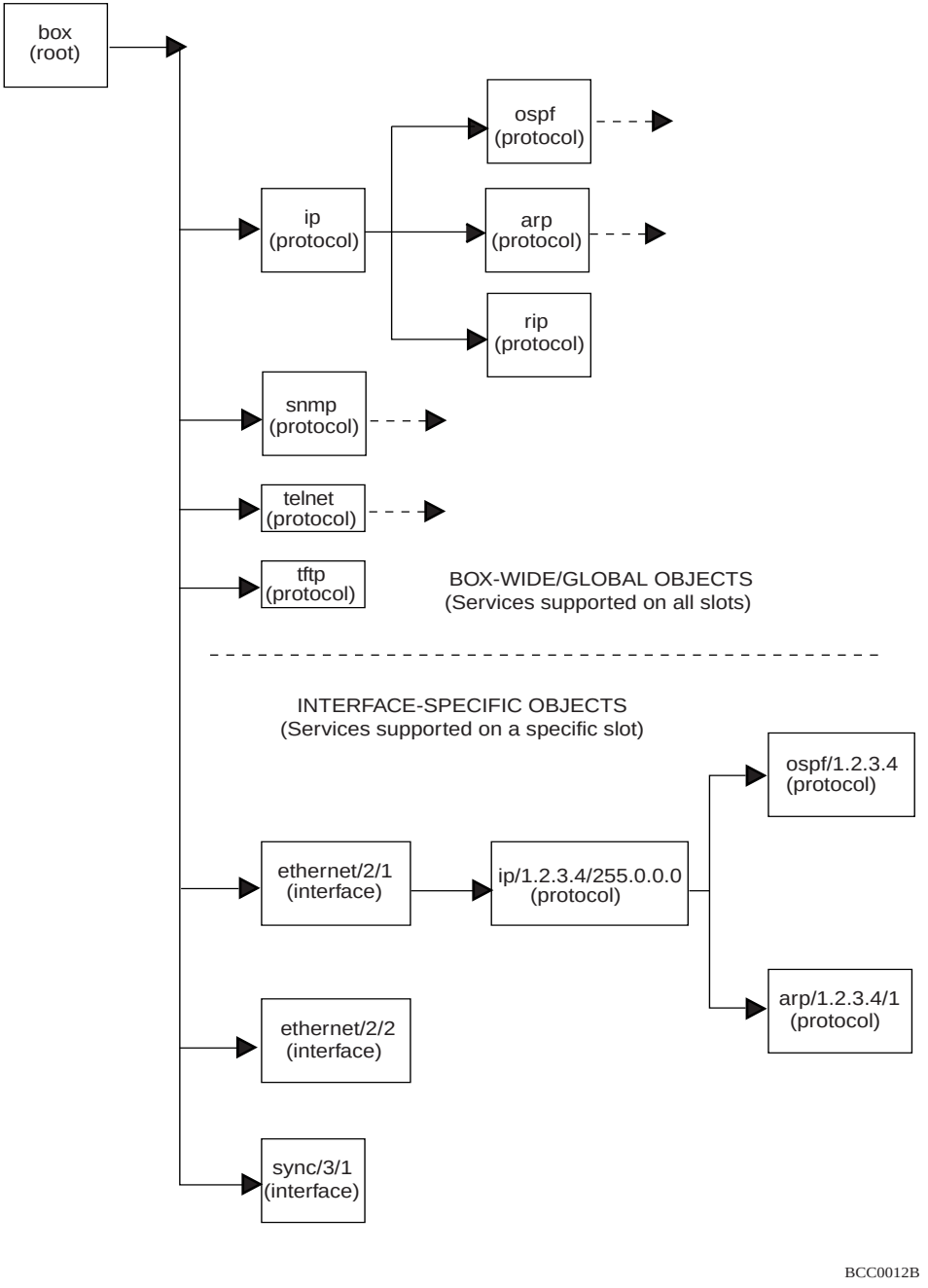


Figure 2-1. Sample BCC Configuration (AN/BN Router)

Note the tree-like hierarchy of this configuration, where

- Box contains global objects IP, Telnet, TFTP, and SNMP, plus the physical interface objects ethernet/2/1, ethernet/2/2, and sync/3/1.
- Global IP contains the ARP, RIP, and OSPF global objects.

Locating Configured Objects

The BCC displays (and understands as command input) the location or *context* of any object such as an IP interface in terms of a path that begins at the root (box) level of the device configuration tree. For example, if you configure an IP interface (address 1.2.3.4, mask 255.0.0.0) on ethernet/2/1, the BCC understands its location as *box; ethernet/2/1; ip/1.2.3.4/255.0.0.0* (Figure 2-2).

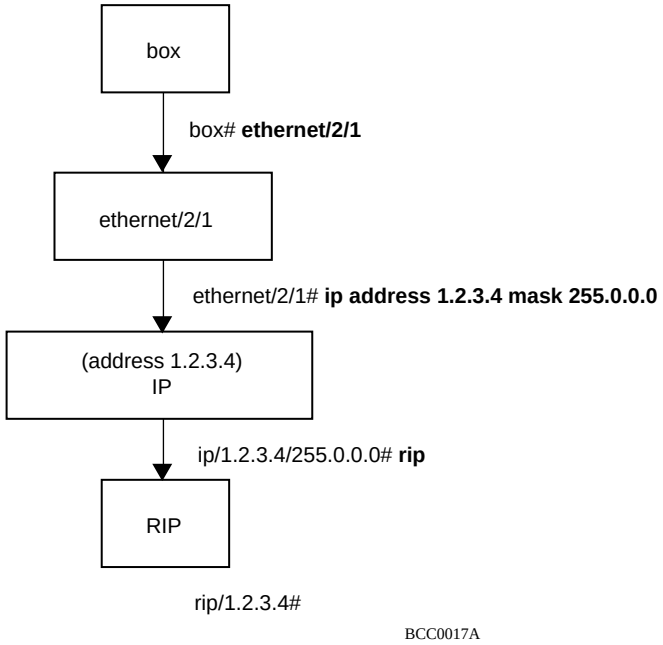


Figure 2-2. Object Location or Context

Each semicolon in the path marks a transition from one level to the next branch level in the device configuration tree. The semicolon is also equivalent to a Return key entered at the end of a command, effectively starting a new command line.

Displaying Context

In BCC configuration mode, use the **pwc** (print working context) command to display the path from root (box) level to your current level in the device configuration tree. For example:

```
rip/192.168.125.34# pwc
```

This command displays the full path from root (box) level to the current level, rip/192.168.125.34. The output appears as follows:

```
box; ethernet/2/1; ip/192.168.125.34/255.255.255.224; rip/192.168.125.34;
```

The **pwc** command displays the BCC instance identifier of each configured object in the path.

Enter the **pwc** command at any prompt in BCC configuration mode.

You can also make a quick determination of your current working location by simply looking at the configuration prompt, which is context-sensitive.

Context-Sensitive Prompts

The BCC shows in the command-line prompt your current working context (location within the BCC configuration tree), for example

```
box# ethernet/2/1
ethernet/2/1# ip/192.168.125.34/255.255.255.224
ip/192.168.125.34/255.255.255.224# rip
rip/192.168.125.34#
```

Note: The prompt contains the BCC instance identifier of the object you specified in the previous command line.

The prompt does not show the *full* path to the current object from root level. The prompt shows only the object that terminates the entire path from root (box) level. To obtain the full path, use the **pwc** command.

Navigating in Configuration Mode

You can navigate from one object to another in BCC configuration mode by using

- The **back** command
- Configuration commands

Navigating with the back Command

In BCC configuration mode, use the **back** command to move a specific number of levels back toward root level. Syntax for the **back** command is as follows:

back [<*n*>]

Entering the **back** command with no argument moves you back one level closer to root level. For example:

```
rip/192.168.125.34# back
ip/192.168.125.34/255.255.255.224# back
ethernet/2/1#
```

Entering the **back** command with an integer moves you from your current working location, back toward root, the number of levels you specify. For example:

```
rip/192.168.125.34# back 2
ethernet/2/1#
```

In this example, the **back 2** command moves you from a current working location of rip/192.168.125.34, back two levels to ethernet/2/1 (with ip/192.168.125.34/255.255.255.224 as the intervening level).

Note: If you enter an integer value that exceeds the actual number of levels back to root (box) level, the BCC returns to root level.

Navigating with Configuration Commands

Using BCC configuration commands, you can

- Move back to a previous context or level.
- Move back to root (box) level.
- Move forward to the next level.
- Move from your current level to any other level in the device configuration tree.

Moving Back One or More Levels

To move from your current working level back one or more levels closer to root (box) level of the device configuration tree, enter the full BCC instance ID of the desired object, as follows:

Example (go back one level)

```
rip/192.168.125.34# ip/192.168.125.34/255.255.255.224
ip/192.168.125.34/255.255.255.224#
```

Example (go back two levels)

```
rip/192.168.155.151#  ethernet/2/1
ethernet/2/1#
```

In the second example, the BCC searches back (toward root) automatically until it finds a context or level where the object you specified (in this case, ethernet/2/1) exists in the router configuration tree. The BCC enters the context of this object, and the command-line prompt displays your new location.

Moving Back to Root (box) Level

You can move back to root level in configuration mode by entering the object name, box.

```
ip/1.2.3.4#  box
box#
```

Moving Forward from Root (box) Level

To move from your current working level to the next configured level (Figure 2-3), enter the BCC instance ID of the desired object, as follows:

Example

```
box#  ethernet/2/1
ethernet/2/1#  ip/1.2.3.4/255.0.0.0
ip/1.2.3.4/255.0.0.0#  rip
rip/1.2.3.4#
```

Notice that a slash character (/) joins the name and any required parameter values to make a BCC instance ID for any configured object.

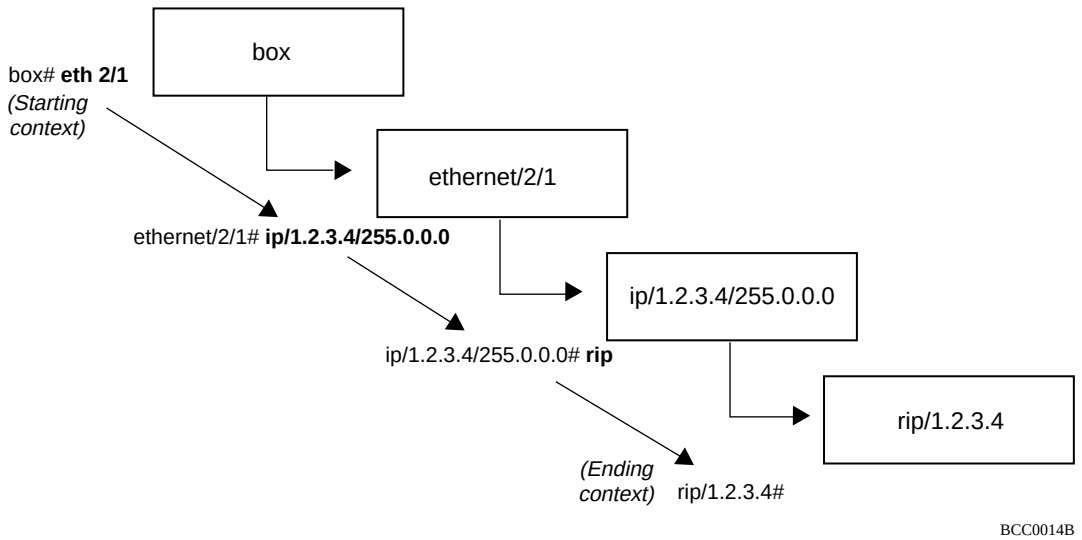


Figure 2-3. Moving Away from Root (box) Level

Specifying a Full Path

You can specify a full, or absolute, path from root (box) level to any configured object. When you enter a path, specify the BCC instance identifier of each object.

Example

Move from ip/192.168.33.66/255.255.255.0 (on ethernet/2/1) to rip/1.2.3.4 on ethernet/2/2 (Figure 2-4).

```
ip/192.168.33.66/255.255.255.0# box;ethernet/2/2;ip/1.2.3.4/255.0.0.0;rip
rip/1.2.3.4#
```

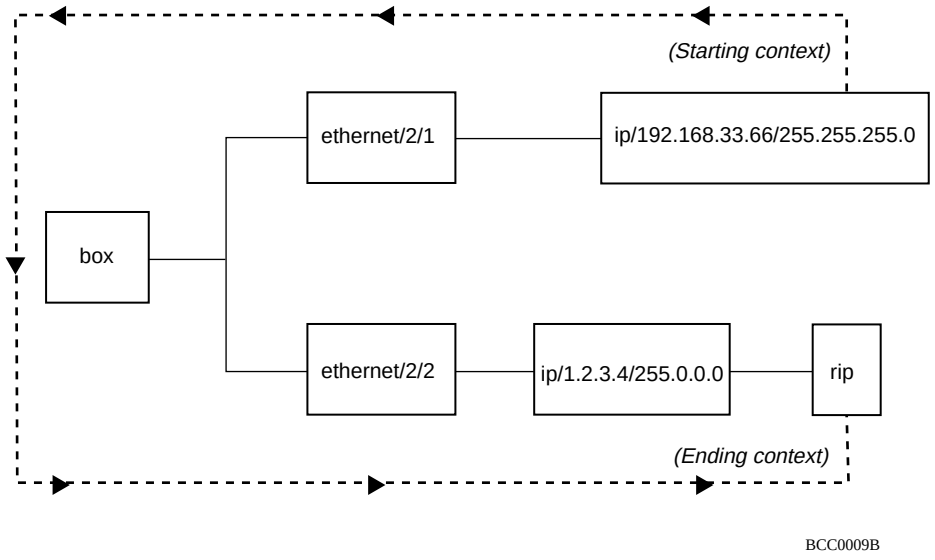


Figure 2-4. Specifying a Full Path

Specifying a Shortened Path

The BCC can automatically search backward (recursively), toward root level, until it finds a level where the object you specify first in the command line exists in the device configuration tree. After the BCC finds that object, it responds to any additional configuration (navigation) commands, starting from that level. This helps to shorten the command line necessary to navigate from one branch level to another.

Example

Move from ip/192.168.33.66/255.255.255.0 on ethernet/2/1 to rip/1.2.3.4 on ethernet/2/2 (Figure 2-5).

```
ip/192.168.33.66/255.255.255.0#  ethernet/2/2;ip/1.2.3.4/255.0.0.0;rip
rip/1.2.3.4#
```

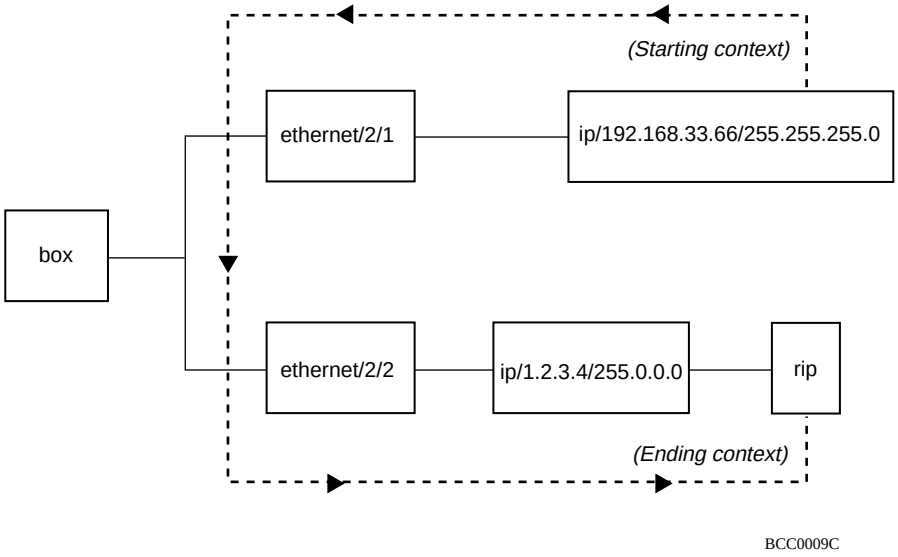


Figure 2-5. Navigating with the BCC Backward (Recursive) Search Feature

In this example, the BCC searches backward to find ethernet/2/2 (specified first in the command line), and then moves sequentially to the other locations (ip/1.2.3.4/255.0.0.0 and rip) specified next in the command line.

Displaying Online Help

The BCC supports the following online Help and Help-oriented features:

Help Feature	Command
Get Help on commands:	
Get an overview of Help-oriented command features.	**help**
Get more detail on Help-oriented command features.	**help -more**
List commands or command options you can enter.	**?** Or: *<command>* **?** Or: *<command> <option>* **?**
Display all BCC commands in alphabetical order (syntax). **Note**: Object and parameter names that you enter as configuration commands are not in this list. (Refer to the **?** and **help tree** commands later in this table.)	**help commands**
Display more detail on all BCC commands in alphabetical order (syntax and terse command descriptions).	**help commands -more**
Get full details on a specific command.	**help** *<command>* Example: To get full details on the **save** command, enter **help save**.
Get Help on symbols used to express BCC command syntax.	**help syntax**
Get configuration data (any BCC context or command mode):	
Show the device configuration tree, as follows: • List, in hierarchical format, all objects configurable in the device configuration tree. • List, in hierarchical format, all objects configurable within the context of a specific box-level *<object>*, where *<object>* is board, console, ip, ftp, snmp, telnet, ftp, tftp, ethernet, tokenring, hssi, sync, fddi, or virtual on an AN or BN router.	 **help tree** **help tree** *<object>*
Show in sourceable BCC syntax the existing configuration of the total device or a configured object you identify by its BCC instance ID.	**show config** [*<BCC_instance_id>*]

Help Feature	Command
Get configuration data (BCC configuration mode only):	
List the names of objects, parameters, and commands you can enter next.	**?**
Get command usage Help and list parameters of an object you can add or modify from your current location.	*<object>* **?**
Get the current value for this parameter of the current object.	*<parameter>*
Get legal, current, and default values for this parameter of the current object.	*<parameter>* **?**

Get legal, current, and default values for this parameter of a configured object you can access from your current working context or level.	*<BCC_instance_ID> <parameter>* **?**
	Example:
	box# **`ethernet/2/1 bofl ?`**
	`Current Value: enable`
	`Legal Value: {enable disable}`
	`Default Value: enable`

List values currently assigned to parameters of this object.	**info**
Describe parameters of the current object.	**help parameters**
Describe parameters of an object outside the current context.	**help parameters *<path>*** (The *<path>* is the sequence of objects between root level and the desired context.) Example: **help parameters ip ospf** Hint: Use the **help tree** command to determine the *<path>* to the desired object.
List other objects currently configured at the next (branch) level of the configuration tree for this device.	**has** **lso**
Get Help on a specific task:	
Get Help on performing common BCC operations.	**help learning-bcc**
Get Help on how to edit BCC commands and command lines.	**help editing**

Example

Get a terse overview of BCC Help features:

```
bcc> help

                                Help Summary
_____________________________________________________________________

In any BCC mode:

  * ?
  * <command> ?
       + Example: show ?
  * <command> <option> ?
       + Example: show ip ?
  * help
  * help -more
  * help <any_command>
       + Example: help pwc
       + Example: help show
  * help commands
  * help commands -more
  * help editing
  * help tree
  * help tree <any_box-level_object>
       + Example: help tree ip
  * show config
  * show config <BCC_instance_ID>
       + Example: show config ip/1.2.3.4/255.0.0.0
  * help syntax
  * help learning-bcc

In BCC configuration mode (HELP-ORIENTED features):

  * ?
  * <object> ?
       + Example: ethernet ?
  * <BCC_instance_ID> ?
       + Example: ip/1.2.3.4/255.0.0.0 ?
      .       .       .
      .       .       .
      .       .       .
  * info
  * lso
  * on
  * pwc
```

Example

Get a more detailed overview of BCC Help features:

```
bcc> help -more

                          System help (any mode):

  * ?
    List all commands you can enter next.
  * <command> ?
    Display syntax/usage help for command options.
    Example: show ?
  * <command> <option> ?
    Display syntax/usage help for command options.
    Example: show ip ?
  * help
    List all BCC on-line help commands.
     .                          .                        .
     .                          .                        .
     .                          .                        .
  ____________________________________________________________________

                            Configuration help:

 (HELP-ORIENTED FEATURES available in BCC configuration mode.)

    * show config
      Show in sourceable BCC syntax everything configured on this
      device.
    * show config [<BCC_instance_ID>]
      Show in sourceable BCC syntax everything configured on this
      object.
      Example: show config ip/1.2.3.4/255/0.0.0
       .                          .                        .
       .                          .                        .
       .                          .                        .
      Display the BCC instance identifier for the parent of the current
      object.
    * pwc
      Display the path to the current object, starting from root (box)
      level of the configuration tree.
```

Example

Get an alphabetical list of BCC system commands (syntax only, with some examples, where needed):

```
bcc> help commands

   SCROLL TO THE COMMAND YOU NEED.
     ____________________________________________________________________

                                Symbols: ? and !

   - - - - - - - - - - - - - - - - -

   (In any mode:)

   ?

   <command> ?

   <command> <option> ?

   !

   ! <repeat_count>

   - - - - - - - - - - - - - - - - -

   In BCC configuration mode:

   <object> ?
   Example (from an ethernet context): ip ?

   <BCC_instance_ID> ?
   Example (from an ethernet context): ip/1.2.3.4/255.0.0.0 ?

   <parameter> ?
   Example: address-resolution ?

   <BCC_instance_ID> <parameter> ?
  Example (from an ethernet context): ip/1.2.3.4/255.0.0.0 address-res ?
     ____________________________________________________________________

        .                          .                          .
        .                          .                          .
        .                          .                          .
```

```
                                    B

back [n]

bcc

boot [{ <vol>:<image_name> | - } { <vol>:<config_name>| - }]
  ──────────────────────────────────────────────────────────────────
     .                          .                          .
     .                          .                          .
     .                          .                          .
  ──────────────────────────────────────────────────────────────────

                                    S

save {config|aliases} <vol>:<filename>

save log [<vol>:<logfile>] [-d<date>] [-t<time>] [-e"<entity>"]
[-f<severity>] [-s<slot ID>]

securelogin

show <entity><option>

source <vol>:<filename>

stamp

system
  ──────────────────────────────────────────────────────────────────

                                    T

telnet [-d] [-e escape_char] [host_ip [port]]

tftp {get|put} <address><vol>:<file_spec>[<vol>:<file_spec>]

type [-x] <vol>:<filename>
  ──────────────────────────────────────────────────────────────────

     .                          .                          .
     .                          .                          .
     .                          .                          .
```

Example

Get a more detailed alphabetized list of BCC commands (with syntax and terse definitions):

```
box#  help commands -more

PAGE OR SCROLL TO THE TOPIC OR COMMAND YOU NEED.
  ____________________________________________________________________
[ <object> ] [ <parameter> ] ?
       List the names of objects, parameters, and commands you can
       enter next. (Configuration mode)

[ <command> ] [ <option> ] ?
       List command options, arguments, and filters. (Any BCC mode)

! [<repeat_count>]
       Repeats the last command you entered, optionally up to the
       number of times you specify in the <repeat_count> argument.
  ____________________________________________________________________
                                    B

back [n]
       Move your current working location or level back n levels
       closer to the root level of the BCC configuration tree. (BCC
       configuration mode only)

bcc

       Start the Bay Command Console (BCC) from the Technician
       Interface prompt. (To exit the BCC, type exit.)

boot [<vol>:<image_name>|- <vol>:<config_name>|-]
       Reboot the system.
  ____________________________________________________________________
                                    C

cd [<vol>:][<directory>]
       Set and/or display the present working volume and directory

clear <sub_commands> <flags>
       Clear specific router information, such as IP, RIP, BGP, and
       EGP routes.
 .                          .                       .
 .                          .                       .
 .                          .                       .
```

Example

Define the purpose of each parameter of ethernet/2/1:

```
ethernet/2/1# help parameters

                          ethernet Parameters:
_____________________________________________________________________

  on
        Identifies the parent(s) of this object.

  state
        Indicates the administrative state of this object (enabled or
        disabled).

  bofl
        Allows breath-of-life polls to be disabled.

  bofl-retries
        Specifies the BOFL Retry Count.

  bofl-timeout
        Specifies the number of seconds for the BOFL timer.

  bofl-tmo-divisor
        Specifies the BOFL TMO divisor.

  circuit-name
        Specifies the circuit Name of this port.

  connector
        -REQUIRED- Specifies the connector associated with this
        interface.

  hardware-filter
        Enables the hardware bridge filter if available.

  receive-queue-length
        Specifies the number of receive buffers dedicated to the chip.

  slot
        -REQUIRED- Specifies the slot associated with this port.

  transmit-queue-length
       Specifies the number of transmit buffers dedicated to the chip.
```

Example

Show all objects you can configure on this device:

```
box#  help tree

                          Router Configuration Tree

  The following hierarchical list shows every object that you can
  configure (but is not necessarily configured at this time) on this
  device. You can use this list as a map to help you navigate within the
  device configuration tree.

  To view only a specific branch of the configuration tree, enter help
  tree <object>, where <object> is any **box-level** object, such as
  ethernet, ip, or telnet.

  ------------------------------------------------------------------------

  * box
       + board
       + ethernet
            o ip
                 # rip
                 # ospf
                      @ neighbor
                 # rdisc
                 # arp
                 # igmp
       + fddi
            o ip
                 # rip
                 # ospf
                      @ neighbor
                 # rdisc
                 # arp
                 # igmp
            .     .    .
            .     .    .
            .     .    .
    + ip
            o ospf
                 # area
                      @ summary
                 # accept
            .     .    .
            .     .    .
            .     .    .
```

Example

List objects you can configure on the ethernet/2/1 branch of the router configuration tree:

```
ethernet/2/1# help tree ethernet

                    Branch Hierarchy for "ethernet"
```

```
* box
     + ethernet
          o ip
               # rip
               # ospf
                    @ neighbor
               # rdisc
               # arp
               # igmp
```

Note: The main difference between the **help tree** and **show config** commands is that **help tree** shows what you *can configure*; **show config** shows you what *has been configured*. For more information about the **show config** command, see "Displaying the Configuration of One Object" on page 2-28.

You can use the output of the **help tree** command to find the configuration or navigation path to a specific object. For example, to configure a global OSPF accept policy, begin in *box* context and configure *ip* (global), then *ospf* within the context of global IP, and then *accept* within the context of global OSPF.

You can also use the output of the **help tree** command to determine the optional <*path*> argument for a **help parameters** command. For example, entering **help tree ip** displays the path to the BGP announce policy object, as follows:

```
ip# help tree ip

                    Branch Hierarchy for "ip"
_____________________________________________________________________

* box
     + ip
          o bgp
               # announce
   .    .    .    .
   .    .    .    .
   .    .    .    .
```

From this data you can obtain parameter descriptions for the BGP announce policy object as follows:

help parameters ip bgp announce

Example

List the names of all objects, parameters, and system commands you can enter from the context of ip/192.168.125.34/255.255.255.0:

```
ip/192.168.125.34/255.255.255.224# ?
Sub-Contexts:
  arp     igmp    ospf    rdisc   rip

Parameters in Current Context:
  address                 cost                    name
  address-resolution      end-station-support     on
  all-subnet-broadcast    has                     proxy
  assocaddr               host-cache-aging        redirects
  broadcast               mask                    state
  cache-size              mask-reply              udp-checksum
  configured-mac-address  mtu-discovery

System Commands:
  ?               delete          mkdir           reset
  attribute       diags           mlist           restart
  back            dinfo           more            rm
  boot            dir             mount           rmdir
  cd              disable         mset            save
  clear           enable          partition       securelogin
  clearlog        exit            password        show
  commit          format          ping            source
  compact         help            pktdump         stamp
  config          history         ***             stop
  context         loadmap         prom            system
  cp              log             pwc             tic
  ***             logout          readexe         unmount
  date            lso             reconfig        xmodem
  debug           mget            record
```

The list of configurable objects (“Sub-Contexts”) are those you can add from your current level (Figure 2-6).

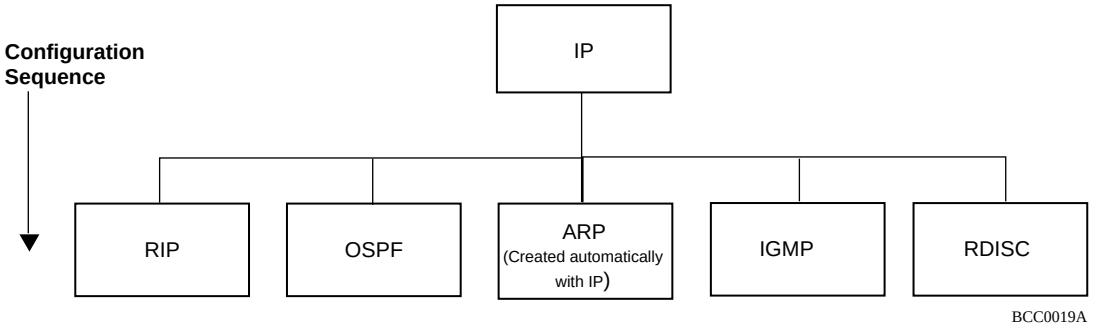


Figure 2-6. Objects to Configure at the Next (Branch) Level

Example

Get the values currently assigned to all parameters of IP (address 1.2.3.4) on ethernet/2/1:

```
ip/1.2.3.4/255.0.0.0# info
  on ethernet/2/1
  state enabled
  address 1.2.3.4
  mask 255.0.0.0
  assocaddr 0.0.0.0
  cost 1
  broadcast 1.2.3.6
  configured-mac-address 0x
  mtu-discovery off
  mask-reply off
  all-subnet-broadcast off
  address-resolution arp
  proxy off
  host-cache-aging cache-off
  udp-checksum on
  end-station-support off
  redirects on
  cache-size 128
```

To get the current value assigned to a specific parameter, enter just the parameter name, as follows:

```
ip/1.2.3.4/255.0.0.0#>  proxy
  proxy off
```

Two parameters, **on** and **has**, have special values within the BCC configuration model.

On -- Identifies the parent of the current object. In the previous example, the ethernet/2/1 object is the parent of the ip/1.2.3.4/255.0.0.0 object. Hence, the value of the **on** parameter for ip/1.2.3.4/255.0.0.0 is ethernet/2/1.

Has -- Just as a directory in a file system often has (or contains) other files, an object in the BCC configuration tree often has (or contains) other objects. For example, in Figure 2-1, the configured object ip/1.2.3.4/255.0.0.0 *has* two other configured objects: arp/1.2.3.4/1 and ospf/1.2.3.4.

Because **on** and **has** are parameters of any object, you can get the current value of **on** or **has** by entering those parameter names at any configuration prompt. The **has** parameter returns no value for leaf objects on the BCC configuration tree. (Leaf objects cannot contain any other objects.)

Examples

Show the parent of ip/1.2.3.4/255.0.0.0:

```
ip/1.2.3.4/255.0.0.0#  on
   ethernet/2/1
```

List other objects that ip/1.2.3.4/255.0.0.0 *has*:

```
ip/1.2.3.4/255.0.0.0#  has
  arp/1.2.3.4/1;  rip/1.2.3.4
```

Using the BCC Help File

To use any BCC command that begins with the verb **help** (for example, **help -more**, **help commands**, or **help tree**), the BCC Help file (called *bcc.help*, by default) must be present on the NVFS default volume. If the file is missing, or if you have not configured the box to find the Help file, the BCC displays the following message:

```
The help data file is missing.  Please transfer “bcc.help” to the default
volume.  You can obtain a copy of “bcc.help” from

* Compact Disk (CD) for this software release. (Released version of the
file.)

* http://support.baynetworks.com/library/tpubs/bcc (Latest version of
the file.)

For more information on BCC help, refer to your Using the Bay Command
Console user guide.
```

If you see this message, first check to see if the Help file already exists on the default volume. If the file is missing, obtain a copy of it and use FTP or TFTP to transfer the file to the default volume of the device. After verifying that the file exists on the default volume, enter BCC configuration mode and enter the following command to specify the name and location of the Help file:

help-file-name [*<vol>* :] {**bcc.help** | *<file_name>* }

Examples

box# **help-file-name 2:bcc.help**
or:
box# **help-file-name 4:new.help**

Note: If you choose a different name for the Help file, be sure to rename the file itself.

Displaying Configuration Data

You can use the **show config** command and the value of the **has** parameter to view Bay Networks device configuration commands and data.

The **show config** command yields command-oriented output for

- The total device configuration
- The configuration of a specific object (such as global IP) on the local device

Displaying the Total Device Configuration

The **show config** command displays the entire device configuration as BCC configuration syntax. You can save the output of the **show config** command as an ASCII file, and then (in configuration mode) source (merge) the contents of that file directly into the active configuration of the same or another device.

The output of **show config** describes

- Existing (configured) objects
- Objects that the BCC automatically added to the active device configuration
- Navigation (**back**) commands necessary to move to the context of the next object configured, or to return to the root (box) level of the active device configuration

Example

```
bcc> show config
box type freln
  board slot 1
      type srml
  back
  board slot 2
      type qenf
  back
  board slot 3
      type wffddi2m
     .          .          .
     .          .          .
     .          .          .
  ethernet slot 2 connector 1
      circuit-name E21-alpha
    ip address 192.168.3.4 mask 255.255.255.224
        broadcast 192.168.3.5
      rip
      back
      arp
      back
    back
  back
     .          .          .
     .          .          .
     .          .          .
  ip
    tcp
    back
    arp
    back
    rip
    back
  back
  ftp
      default-volume 2
  back
  snmp
    community label public
        access read-write
      manager address 0.0.0.0
      back
      manager address 192.168.9.9
     .          .          .
     .          .          .
     .          .          .
```

Displaying the Configuration of One Object

You can view objects configured within a specific part of the BCC configuration tree by using the **show config** command, the **lso** command, or the value of the **has** parameter.

Example 1 (show config)

You can use the **show config** command to view the configuration of a specific configured object from any BCC prompt. You must enter the full BCC instance identifier of the object you want to see. For example, to see everything configured on ethernet/2/1, enter the following command from any BCC prompt:

```
bcc>  show config ethernet/2/1
ethernet slot 2 connector 1
    circuit-name E21-alpha
  ip address 1.2.3.4 mask 255.255.255.224
      broadcast 1.2.3.6
    rip
    back
    arp
    back
  back
back
```

Refer to "Terminology and Concepts" in Chapter 1 for more information about BCC instance identifiers.

Example 2 (has or *lso)*

You can enter **has** or **lso** in configuration mode to view any instances of objects configured at your current location in the active device configuration, as follows:

```
ip/1.2.3.4/255.255.255.224# has
  has rip/1.2.3.4 arp/1.2.3.4/1

ip/192.168.155.151/255.255.255.0#  lso
  arp/192.168.155.151/1 ; rip/192.168.155.151
```

Use the **lso** command or the **has** parameter if you have no need to display the contents of the current object in full BCC configuration syntax.

Displaying Binary Configuration Files as BCC Syntax

After booting the device from a binary configuration file, you can

1. Use the **show config** command to view the current device configuration in readable BCC syntax. For an example of **show config** output, refer to "Displaying the Total Device Configuration" on page 2-26.
2. Enter new configuration commands to override elements of the active device configuration. (See Chapter 3 for more information about how to enter BCC configuration commands.)
3. Save the file as an ASCII configuration file that the BCC can read using the **source** command. (See "Saving Commands on a Workstation" on page 3-14 for instructions.)
4. Save the file as a binary configuration file, bootable on the same device or on another device. (See "Saving the Active Configuration to an NVFS Volume" on page 3-14 for instructions.)

Chapter 3
Entering Commands

You can enter BCC configuration commands using basic (full), default, or abbreviated syntax. In addition, *command operators* (also called *methods*) enable you to perform certain operations more efficiently from your current location in the device configuration tree.

This chapter provides information about the following topics:

Topic	Page
Command Input Features	3-2
System Commands	3-6
Configuration Commands	3-7
Command Operators	3-11
Editing Command Lines	3-13
Saving Commands on a Workstation	3-14
Saving the Active Configuration to an NVFS Volume	3-14

Command Input Features

This section contains information about

- Using Command Abbreviations
- Recalling Commands
- Entering Commands from a File
- Entering Multiple Commands per Line
- Continuing a Command Line
- Entering Comments

Using Command Abbreviations

For configuration command input, you can shorten existing object and parameter names (for example, eth = ethernet). You must enter a sufficient number of characters for the BCC to recognize that name uniquely. This is the BCC "minimum to distinguish" feature, which applies only when entering object or parameter names in configuration mode.

Example

```
box#  eth 5/1
ethernet/5/1#  back
box#  tf
tftp#
```

The BCC does not recognize abbreviated system (nonconfiguration) commands. For example, the BCC does not recognize **sh** as **show**.

Recalling Commands

The BCC supports a configurable command history buffer, from which you can recall commands recently entered. Recall/reenter commands from the history list as follows:

Purpose	Command
Recall the previous command from the history list.	Up arrow key or Control + p
Recall the next command from the history list.	Down arrow key or Control + n

The command history list contains up to 20 commands by default. You can increase the number of commands in the history list to a maximum of 40 by setting new values for the history parameter of the console and telnet objects.

Example

```
box# telnet
telnet# server
server# history 30
server# history
   history 30
server# back
telnet# back
box#
```

Entering Commands from a File

With Manager login privileges, you can use the **source** command in configuration mode to read BCC configuration and navigation commands from a designated ASCII source file into the active device configuration.

Caution: The **source** command makes immediate changes to the active device configuration.

The **source** command *merges* new configuration data from a file with existing data in device memory. If the file you specify contains configuration commands pertaining to objects already defined on the device, those commands dynamically overwrite the current configuration.

Note: If the BCC detects an error in the source file, it stops reading commands into the device configuration.

Syntax for the **source** command is as follows:

source *<volume>*:*<filename>*

For information about how to save as an ASCII file the output of the **show config** command, or a manually entered set of commands, refer to "Saving Commands on a Workstation" on page 3-14.

Note: If you edit ASCII files containing BCC commands, you must adhere to BCC syntax requirements, including any commands necessary to navigate to each level of the device configuration tree.

Entering Multiple Commands per Line

To enter multiple commands in the same line, type a semicolon (;) wherever you would press Return to terminate a command. For example, to configure RIP on ip/1.2.3.4/255.0.0.0 on ethernet/2/1, enter

```
box#  ethernet/2/1;ip 1.2.3.4/255.0.0.0;rip
rip/1.2.3.4#
```

Continuing a Command Line

You can continue a command line by entering a backslash (\ ) character at the end of the current text line. The BCC treats characters on the next physical line as part of the same BCC logical command line.

You must immediately follow the initial escape character (\ ) with a newline (Return) character. The BCC treats these two characters and any trailing spaces as if they were exactly one space. Until you press Return without a preceding escape (\ ) character, the BCC replaces the # symbol in the context-sensitive prompt with an underscore (_) character.

Example

```
ip/1.2.3.4/255.0.0.0# cost 2 \
ip/1.2.3.4/255.0.0.0_ mask-reply on \
ip/1.2.3.4/255.0.0.0_ proxy on \
ip/1.2.3.4/255.0.0.0_ aging cache-on
ip/1.2.3.4/255.0.0.0#
```

Some command symbols normally used in pairs to denote the beginning and the end of a set of data also produce the continuation (underscore) prompt, including braces ({ }), brackets ([ ]), and quotes (“ ”).

For example:

```
box#  { ...
box_  ...}
box#
```

or:

```
box#  [...
box_  ...]
box#
```

or:

```
box#  “...
box_  ...”
box#
```

The BCC displays the underscore prompt after you type the opening symbol because it is expecting data plus the corresponding closing symbol.

Note: If you inadvertently type one of the opening symbols and see an underscore prompt, just type the corresponding closing symbol to restore the normal (#) prompt in BCC configuration mode.

Entering Comments

You can use a text editor (such as *vi* on a UNIX workstation) to add descriptive comments to a BCC configuration file. Enter comments in the following format:

```
box# <command> ;# comment
```

or

```
box# #comment
box# <command>
```

Example

```
box#  board slot 1 type andse;# 192.168.47.129  192.168.47.21
```

When you finish editing the file, save it on your workstation or PC. The comments are for reference only.

The BCC does not use or retain any comments you enter

- During a BCC configuration session with a Bay Networks device
- When sourcing a configuration file that contains comments (refer to "Saving Commands on a Workstation," later in this chapter)

For this reason, comments also do not appear in the output of any **show config** command.

System Commands

AN and BN (BLN/BCN) routers support all system commands described in Appendix B.

For more detailed information about a specific command:

- Enter **help** <*command*> at any BCC prompt (for example, **help save**).
- Refer to the guide *Using Technician Interface Software.*

Configuration Commands

Caution: Configuration commands and **source** commands make real-time changes to the device configuration.

This section describes how to enter BCC configuration commands using any of the following formats:

- Basic (full) syntax
- Default syntax
- Abbreviated syntax

BCC syntax consists of object names, parameter names and values, and various types of punctuation. Note in particular that

- All object and parameter names appear as one word (hyphenated where necessary) in the BCC command line.
- Parameters have either a single value or multiple values enclosed in braces {x y z} in the command line. You either accept the default value or supply a value for each parameter associated with a configurable object.
- Parameters and their values must appear as a pair in the same command line.

Using Basic (Full) Syntax

The basic or full syntax for BCC commands consists of the following required and optional elements:

{*<object-name>*} {*<required_parameter> <value>* ... } ...
<parameter> <value> ... <parameter> <value>

The BCC requires input for any elements enclosed by braces ({ }).

{<*object-name*>} is either

- The name of an object you want to configure
- The BCC instance identifier of an object already configured

The BCC assumes that an object you specify is new (and will create it) if it is not in the current configuration. If an object you specify already exists in the current configuration, the BCC assumes that you want to modify that object.

{<*required_parameter*> <*value*>} is any parameter-value pair required to uniquely identify an object you specify in a BCC command line. An object may have one or more required parameters.

For example, the full syntax for configuring an Ethernet interface is

box# **ethernet slot** ***<slot_no.>*** **connector** ***<connector_no.>***

Note: You cannot change the value of a parameter that makes up the BCC instance identifier of a configured object. For example, you cannot modify the address value assigned to an IP interface previously configured. (Address is a required parameter for configuring an IP interface.) To change the value of a required parameter, you must delete the object, then add it back into the device configuration with new required values.

[<*parameter*> <*value*>] is any parameter-value pair you can optionally customize for an object you specify in a BCC command line.

Using Default Syntax

Using default syntax, you do not have to enter the name of a required parameter; you enter only its value at the proper location in the command line.

Example

ethernet <*slot*>**/**<*connector*>

With values:

ethernet 2/1

If you enter the name of an object without values for its required parameters, or with values inappropriate for its required parameters, the BCC returns usage Help, as shown in the following example:

```
box# ethernet
Required attribute "slot" was not specified for class: Ethernet.

Usage: "ethernet slot <value> connector <value>"
Or:    "ethernet <slot>/<connector>"
```

The following commands are equivalent.

Using full syntax:

```
box#  ethernet slot 2 connector 1
ethernet/2/1#
```

Using default (minimum) syntax:

```
box#  ethernet 2/1
ethernet/2/1#
```

ethernet is the object you want to configure.

slot and **connector** are the required parameters of **ethernet**.

The ability to configure objects using only the values for required parameters is the “minimum to configure” feature of the BCC. Typically, the BCC requires you to enter the values but not the names of required parameters. In rare cases, however, the BCC requires you to explicitly enter the name of a required parameter along with its value. (The usage Help indicates the command elements you need to enter.)

Using Abbreviated Syntax

You can abbreviate BCC commands in the following manner:

```
box# eth 2/1
```

This command is the same as either of the following two commands:

```
box# ethernet slot 2 connector 1
```

```
box# ethernet 2/1
```

The BCC completes or expands abbreviated object and parameter names when you press Return.

If you press Return before entering a sufficient number of characters for the BCC to recognize the name of the object or parameter you want to configure, the BCC returns an error message:

```
box# invalid command name " <string> "
```

Specifying Parameter Values

You must specify each parameter value in the form of a parameter-value pair. Each pair is a command argument pertaining to the object named first in the command line.

For example, the following command changes the BOFL timeout interval from 5 seconds to 4 seconds on ethernet/1/1:

```
box#  ethernet/1/1 bofl-timeout 4
```

ethernet is the object type, and **1/1** represents the specific slot and connector location for this Ethernet port.

bofl-timeout is the Ethernet interface parameter you want to modify, and **4** is the new value for the **bofl-timeout** parameter.

bofl-timeout 4 is the parameter-value pair.

Required, Derived, and Other Parameters

The BCC indicates when parameter values are required (you must supply a value) or derived (the BCC supplies a value). For all other parameters, the BCC supplies a user-modifiable default value.

Note: The BCC uses required parameters to make up the unique instance identifier for a configured object. You cannot change the value of any parameter that makes up the instance identifier for an object. To change these parameter values, you must delete the object, and then add it back into the device configuration with new values.

Specifying Multiple Parameter-Value Pairs

Within a specific context in configuration mode, you can

- Enter an object name and one parameter-value pair per command line.
- Enter an object name and multiple parameter-value pairs (each pair separated by a space) on the same command line.

Example (one parameter-value pair per line)

```
box#  ethernet 2/1
ethernet 2/1#  bofl-retries 6
ethernet 2/1#  bofl-timeout 7
ethernet 2/1#  hardware-filter enabled
ethernet 2/1#
```

Example (multiple parameter-value pairs per line)

```
ethernet 2/1#  ip address 1.2.3.4 mask 255.255.255.0 redirects off
ip.1.2.3.4/255.255.255.0#  ospf area 2.3.4.54 hello-interval 5
ospf/1.2.3.4#
```

Command Operators

BCC command operators (also called *methods*) perform a named operation within the current or specified configuration context. AN/BN routers support the following command operators in BCC configuration mode with read-write privileges (Table 3-1).

Table 3-1. BCC Command Operators (Methods)

Operator	Function
disable	Change the administrative state of a configured object from enabled to disabled, as follows: `ip/1.2.3.4/255.0.0.0# disable` You can accomplish the same change by assigning the value “disabled” to the state parameter of an object you want to disable.
enable	Change the administrative state of a configured object from disabled to enabled, as follows: `ip/1.2.3.4/255.0.0.0# enable` You can accomplish the same change by assigning the value “enabled” to the state parameter of an object you want to reenable.
delete	Delete the object identified in the BCC context-sensitive prompt. For example, the following command deletes the IP interface (address 1.2.3.4): `ip/1.2.3.4/255.0.0.0# delete` **CAUTION:** Deleting an object at one level of the configuration tree causes the BCC to automatically delete any branches configured on that object. For example, if you delete an IP interface, the BCC deletes any protocols (such as RIP, ARP, or OSPF) configured on that interface.

info	List the names and values currently assigned to parameters of the current working context. For example, entering **info** after **ip** (at box level) lists the values currently assigned to parameters of the IP global object:
	`ip# info`
	`on box`
	`state enabled`
	`forwarding forwarding`
	`ttl 30`
	`cache-timeout default`
	`mib-table route`
	`all-subnets disabled`
	`classless disabled`
	`max-policies 32`
	`route-filters enabled`
	`rip-max-paths 1`
	`ecmp-method disabled`
	`isp-mode disabled`
	`ospf-max-paths 1`
	`icmp-error-limit 0`

Editing Command Lines

Table 3-2 describes the keystrokes you can use to edit BCC command lines.

Table 3-2. Keystrokes for Editing BCC Command Lines

Editing Function	Keystrokes
Move the cursor left	CONTROL + b or left arrow key
Move the cursor right	CONTROL + f or right arrow key
Delete the current line	CONTROL + u
Delete the word at the cursor location	CONTROL + w
Delete the character at the cursor location	CONTROL + d
Move the cursor to the beginning of the line	CONTROL + a
Move the cursor to the end of the line	CONTROL + e
Toggle insert mode	CONTROL + o
Delete previous character	BKSP or DEL, or CONTROL + h
Interrupt	CONTROL + c
Start echo to the screen	CONTROL + q
Stop echo to the screen	CONTROL + s
Recall previous command	CONTROL + p or up arrow key
Recall next command	CONTROL + n or down arrow key

For example, use the up arrow key (or CONTROL + p) to review your last input, and then use other control-key combinations to edit the command line as needed.

Saving Commands on a Workstation

If you log in to an AN/BN router from a workstation using Telnet or terminal emulation, you can use the native capabilities of the workstation to

- Save the output of any **show config** command from the screen to an ASCII file.
- Save a sequence of manually entered BCC commands to an ASCII file.

You can also use an ASCII text editor on a workstation to create a file from which the BCC can read configuration and system commands.

Note: If you create or edit ASCII files containing BCC commands, you must adhere to the BCC syntax requirements described in this guide. For example, you must include any commands necessary to navigate to each level of the device configuration tree, and any commands necessary to add, modify, or delete objects in the device configuration.

By saving BCC commands to an ASCII file, you can subsequently

- Edit the commands offline.
- Add comments to the file to describe details of the configuration (refer to "Entering Comments," earlier in this chapter).
- Save the edited file for later use.
- Use the BCC **source** command to edit the active configuration of Bay Networks devices of the same type and hardware configuration. (See "Entering Configuration Commands from a File" on page 4-16).

Saving the Active Configuration to an NVFS Volume

When you finish using BCC commands to modify an existing configuration, save the new configuration to a file on an NVFS (flash) volume. (At boot time, the router loses any configuration changes not previously saved to a NVFS volume.) The following command saves *config* as a bootable binary file on a volume you specify:

```
bcc>  save config <volume>:<filename>
```

Chapter 4
Configuring a Network Device

This chapter describes how to complete the following tasks using BCC commands:

Topic	Page
Creating and Modifying a Device Configuration	4-1
Entering Configuration Commands from a File	4-16
Disabling a Configured Object	4-17
Enabling a Configured Object	4-18
Deleting a Configured Object	4-19
Configuration Command Responses	4-20

Creating and Modifying a Device Configuration

You configure a Bay Networks device by defining a set of objects that collectively define its behavior on a network. Each object has a set of parameters with values set either by you or by the device software. You configure a device starting at the root (box) level of the device configuration tree.

Following is a BCC configuration sequence that applies to a variety of network devices.

1. Open a Technician Interface session and start the BCC interface.
2. Start BCC configuration mode.
3. Use BCC configuration commands to create (add) new objects in the device configuration, and to modify default values for parameters of each object to meet the requirements of your network. Be sure to enable any box-wide protocols not enabled automatically by the BCC software, for example, TFTP and Telnet Server.

4. Use the **save** command to save your configuration as a bootable (binary) file on the device.
5. Exit BCC configuration mode, exit the BCC to the Technician Interface, and log out of the device.

You may find it helpful to first diagram what you want to configure in terms of the BCC configuration tree or hierarchy for the device. For example, refer to the following sample router configuration.

Sample Router Configuration

The following example shows a sequence of commands you can use to configure a BCN router on a network. Assumptions for this example are that you first complete the physical installation of the router, then boot the router using the image (*bn.exe*) and the minimum configuration file (*ti.cfg)*.

This example creates the following objects in the total router configuration (Figure 4-1):

- IP (global)
  - -- ARP (global) on IP
  - -- RIP (global) on IP
- SNMP (global)
  - -- Community “public” on SNMP
    - Manager (address 0.0.0.0) on Community “public”
- FTP (global)
- TFTP (global)
- Telnet (global)
  - -- Server (global) on Telnet
- Quad Ethernet interface in slot 13
  - -- IP interface (address 192.168.133.114) on Ethernet connector 1
    - ARP on IP interface 192.168.133.114
    - RIP on IP interface 192.168.133.114

- Synchronous interface in slot 5
- Dual Token Ring interface in slot 9
- FDDI interface in slot 11
- SRML system controller in slot 7

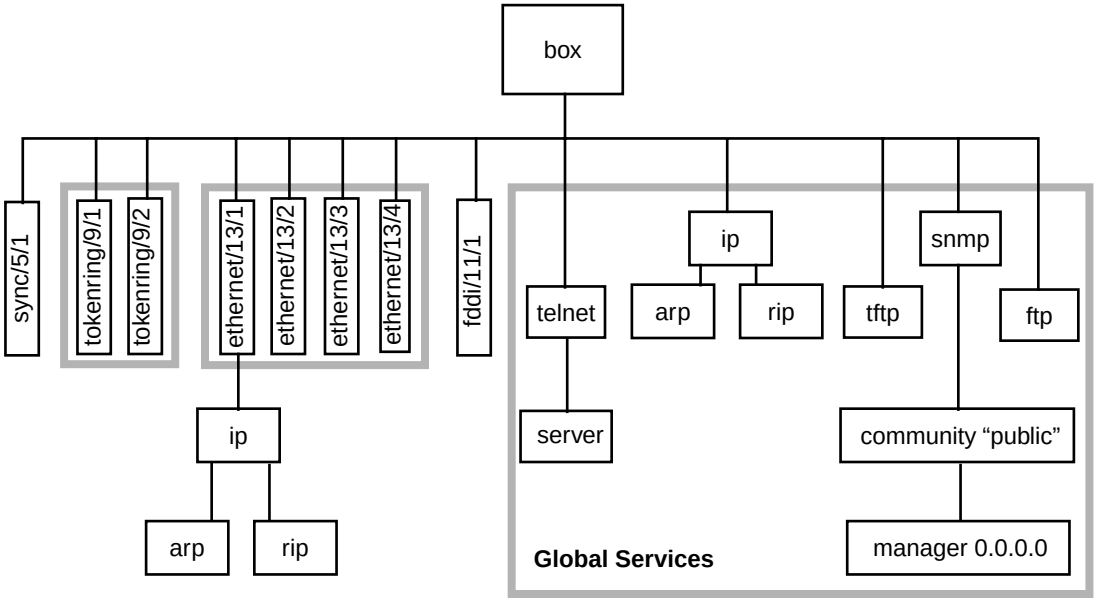


Figure 4-1. Sample BCC Configuration (BCN Router)

After you create a diagram of the device configuration tree, configure the device using a cycle of BCC configuration commands similar to those shown in Figure 4-2.

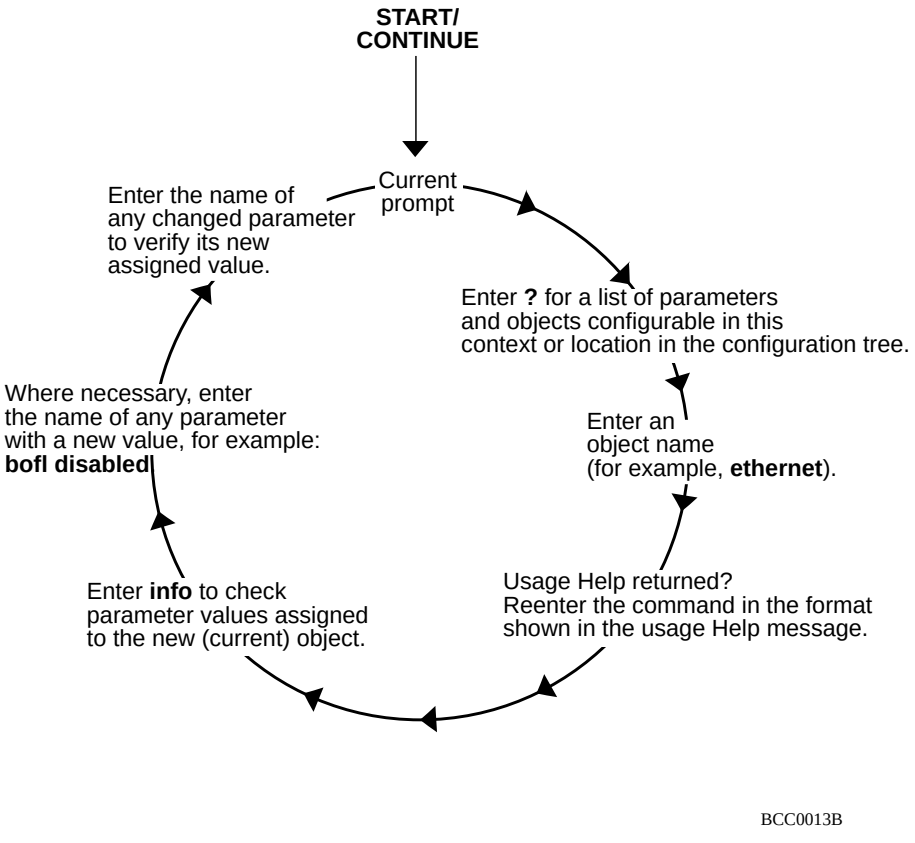


Figure 4-2. Typical BCC Configuration Cycle

To create the sample configuration shown in Figure 4-1 using BCC commands, proceed as follows:

1. **Log on to the router as Manager, so that you can modify the device configuration.**

```
Bay Networks, Inc. and its Licensors.
Copyright 1992, 1993, 1994, 1995, 1996, 1997. All rights reserved.

Login: Manager
Mounting new volume...
Device label:
Directory: 2:
New Present Working Directory: 2:

      Welcome to the Backbone Technician Interface

Router1>
```

2. **Start the BCC by entering bcc at the Technician Interface prompt.**

```
Router1> bcc

     Welcome to Bay Command Console!

     * To enter configuration mode, type config

     * To list all system commands, type ?

     * To exit the BCC, type exit

bcc>
```

3. **Enter BCC configuration mode by entering config.**

```
bcc>  config
box#
```

4. Check the hardware configuration of the router.

```
box# show config

box type frecn (BCN router with a FRE-2 controller)
  board slot 5
      type sync (Synchronous link module in slot 5)
  back
  board slot 7
      type srml (System resource module in slot 7)
  back
  board slot 9
      type dtok (Dual token ring link module in slot 9)
  back
  board slot 11
      type wffddi2m (Multimode FDDI link module in slot 11)
  back
  board slot 13
      type qenf (Quad Ethernet with filters in slot 13)
  back
  console portnum 1 (console device on port 1)
      prompt {"%slot%:"}
      auto-manager-script automgr.bat
      auto-user-script autouser.bat
  back
back
box#
```

Note how output of the **show config** command automatically includes navigation (**back**) commands. If you save this output to a file, you can reenter them automatically using the BCC **source** command. (See "Entering Commands from a File" in Chapter 3.)

5. Choose a port (interface type, slot, and connector) for the initial IP interface to the router.

```
box# ethernet slot 13 connector 1
ethernet/13/1#
```

6. Check to see what you can configure (sub-contexts and parameters) at this level.

```
ethernet/13/1# ?
Sub-Contexts:
  ip

Parameters in Current Context:
  bofl                    circuit-name            on
  bofl-number             connector               receive-queue-length
  bofl-retries            hardware-filter         slot
  bofl-timeout            has                     state
  bofl-tmo-divisor        name                    transmit-queue-length

System Commands:
  ?               delete          mkdir           reset
  attribute       diags           mlist           restart
  back            dinfo           more            rm
  boot            dir             mount           rmdir
  cd              disable         mset            save
  clear           enable          partition       securelogin
  clearlog        exit            password        show
  commit          format          ping            source
  compact         help            pktdump         stamp
  config          history         pop             stop
  context         loadmap         prom            system
  cp              log             pwc             tic
  cwc             logout          readexe         unmount
  date            lso             reconfig        xmodem
  debug           mget            record
ethernet/13/1#
```

Based on the “Sub-Contexts” list, you can add **ip** (interface IP) to this Ethernet interface. At this level, the list of configurable objects depends on the *board* type described in the current prompt. For example, 100BASE-T modules additionally allow you to configure an “autoneg” (autonegotiation) object at this level.

You can also configure (modify) values currently assigned to “Parameters in Current Context” (parameters of ethernet/13/1). The list of “System Commands” always appears along with information about any configurable objects and parameters.

7. Add IP (address 192.168.133.114) to ethernet/13/1.

```
ethernet/13/1# ip 192.168.133.114
Usage: "ip address <value> mask <value>"
Or:    "ip <address>/<mask>"
Required attribute "mask" was not specified for class: InterfaceIp.
```

The error message appears because the BCC requires you to enter a mask value whenever you create an IP interface. Because the first octet of the address is 192 (0xC0), this is a Class C address, requiring the first 3 octets to be the network portion of the interface address. You can express the corresponding mask value as either 255.255.255.0 (using dotted-decimal notation) or 24 (the number of bits making up the network portion of the IP interface address), as follows:

```
ethernet/13/1# ip 192.168.133.114/24
ip/192.168.133.114/255.255.255.0#
```

The BCC converts the integer representing the number of mask bits into a dotted-decimal mask value, as shown in the last prompt.

8. Check the values currently assigned to parameters of IP on this interface.

```
ip/192.168.133.114/255.255.255.0# info
  on ethernet/13/1
  state enabled
  address 192.168.133.114
  mask 255.255.255.0
  assocaddr 0.0.0.0
  cost 1
  broadcast 0.0.0.0
  configured-mac-address 0x
  mtu-discovery off
  mask-reply off
  all-subnet-broadcast off
  address-resolution arp
  proxy off
  host-cache-aging cache-off
  udp-checksum on
  end-station-support off
  redirects on
  cache-size 128
```

9. Change the subnet mask to 255.255.255.224.

```
ip/192.168.133.114/255.255.255.0# mask 255.255.255.224
"mask" attribute may not be modified
ip/192.168.133.114/255.255.255.0#
```

You cannot modify any parameter values included in the BCC instance identifier of a configured object, in this case, ip/192.168.133.114/255.255.255.0. To change the value of any parameter that is part of a BCC instance ID, you must first delete the object, then re-create it with the desired parameter values. In this case, you must delete the IP interface and then re-create it on ethernet/13/1, using the mask value of 255.255.255.224, as follows:

```
ip/192.168.133.114/255.255.255.0# delete
ethernet/13/1# ip 192.168.133.114/255.255.255.224
ip/192.168.133.114/255.255.255.224#
```

10. Check to see what you can configure at this level.

```
ip/192.168.133.114/255.255.255.224# ?
Sub-Contexts:
  arp     igmp    ospf    rdisc   rip

Parameters in Current Context:
  address                 cost                    name
  address-resolution      end-station-support     on
  all-subnet-broadcast    has                     proxy
  assocaddr               host-cache-aging        redirects
  broadcast               mask                    state
  cache-size              mask-reply              udp-checksum
  configured-mac-address  mtu-discovery

System Commands:
  .       .       .
  .       .       .
  .       .       .
```

You can modify values currently assigned to parameters of ip/192.168.133.114/255.255.255.224, or you can add RIP, OSPF, Router Discovery, or ARP to this interface.

11. Add RIP as the routing protocol (by default, RIP1) on this interface.

```
ip/192.168.133.114/255.255.255.224# rip
rip/192.168.133.114#
```

12. Return to root (box) level to configure global system services.

```
rip/192.168.133.114# box
box#
```

13. Check to see what global services (protocols) and interfaces you can configure at this level.

```
box# ?
Sub-Contexts:
  board           ftp             snmp            tokenring
  console         hssi            sync            virtual
  ethernet        ip              telnet
  fddi            ntp             tftp

Parameters in Current Context:
  build-date      has             name            uptime
  build-version   help-file-name  on
  contact         location        system-name
  description     mib-counters    type

System Commands:
.       .       .
.       .       .
.        .       .
```

You can add any of the following global services (affecting all slots): IP, FTP, NTP, SNMP, Telnet, and TFTP.

You can add any of the following interfaces: Ethernet, FDDI, HSSI, synchronous, token ring, or Virtual.

You can view the configuration of a board in any slot, but you cannot modify the parameters of any board object.

14. List the objects already configured at box level.

```
box# lso
board/11        board/5         board/9         ethernet/13/1
board/13        board/7         console/1       ip
box#
```

When you added the first instance of IP to the box (ip/192.168.133.114/255.255.255.224), the BCC automatically created the global IP object at box level.

15. Add SNMP to the device.

```
box# snmp
snmp#
```

16. Check to see what you can configure next at this level.

```
snmp# ?
Sub-Contexts:
  community       trap-entity     trap-event

Parameters in Current Context:
  authentication-traps    lock-timeout            state
  has                     name                    type-of-service
  lock                    on
  lock-address            scope-delimiter

System Commands:
.       .       .
.       .       .
.        .       .
```

You can modify values currently assigned to attributes of SNMP, and you can add a community, define a trap entity, or define a trap event.

17. View the parameter definitions available using the BCC help parameters command.

```
snmp# help parameters

                              snmp Parameters:
   ____________________________________________________________________

   on
          Identifies the parent(s) of this object.

   state
          Indicates the administrative state of this object (enabled or
          disabled).

   authentication-traps
          Sends trap for sets from false Mgr or Community.

   lock
          Allows the locking mechanism to be disabled.

   lock-address
          Allows the lock address to be cleared.

   lock-timeout
        .          .          .
        .          .          .
        .          .          .
```

18. Define the SNMP community "public."

```
snmp# community public
community/public#
```

19. Check the values currently assigned to parameters of this SNMP community.

```
community/public# info
  on snmp
  label public
  access read-only
  scope-type {}
```

20. To allow network management applications (such as Site Manager) to modify the device configuration, modify the value of the access attribute to read-write.

```
community/public# access read-write
community/public#
```

21. Define an SNMP manager for the router.

```
community/public# manager
Usage: "manager address <value>"
Or:    "manager <address>"
Required attribute "address" was not specified for class: SnmpManager.
```

The BCC error message indicates what you left out and automatically provides extended usage Help on how to configure an SNMP manager.

22. Try again to add the manager, this time supplying a value for its required attribute (parameter), address. (You must enter a value but not the name for a required parameter.)

```
community/public# manager 0.0.0.0
manager/public/0.0.0.0#
```

23. Enable the Telnet server entity on the router.

```
manager/public/0.0.0.0# telnet
telnet#  server
server#
```

24. Add TFTP services globally to the router.

```
server# tftp
tftp#
```

The BCC automatically searches back (toward root) to find the parent context suitable for TFTP (in this case, box). The BCC then adds TFTP to the device configuration. Note the new (`tftp#`) prompt.

25. Check values currently assigned to attributes of TFTP.

```
tftp# info
  on box
  state enabled
  default-volume 2
  retry-timeout 5
  close-timeout 25
  retry-count 5
```

26. Change the default volume number for TFTP to 5.

```
tftp# def 5
tftp#
```

27. Verify the change to the default-volume number.

```
tftp# def
  default-volume 5
```

28. Add FTP globally to the router.

```
tftp# ftp
ftp#
```

The BCC automatically searches back (toward root) to find the parent context suitable for FTP (in this case, box). The BCC then adds FTP to the device configuration. Note the new (ftp#) prompt.

29. Check the definitions for parameters of FTP.

```
ftp# help param
                              ftp Parameters:
     ____________________________________________________________________

   on
          Identifies the parent(s) of this object.

   state
          Indicates the administrative state of this object (enabled or
          disabled).

   default-volume
          Specifies the default volume where transferred files are
          written/retrieved.

     .       .       .
     .       .       .
     .       .       .
```

30. Check values currently assigned to parameters of FTP.

```
ftp# info
  on box
  state enabled
  default-volume 2
  login-retries 3
  idle-timeout 900
  max-sessions 3
  tcp-window-size 60000
```

31. Change the default volume number to 5.

```
ftp# def 5
ftp#
```

32. Verify the change to the default volume number.

```
ftp# def
  default-volume 5
```

33. Recheck the total device configuration.

```
ftp# show config
box type frecn
  board slot 5
      type sync
  back
  board slot 7
      type srml
  back
  board slot 9
      type dtok
  back
  board slot 11
      type wffddi2m
  back
  board slot 13
      type qenf
  back
  console portnum 1
      prompt {"%slot%:"}
      auto-manager-script automgr.bat
      auto-user-script autouser.bat
  back
  ethernet slot 13 connector 1
      circuit-name E131
    ip address 192.168.133.114 mask 255.255.255.224
      arp
      back
```

```
      rip
      back
    back
  back
  ip
    arp
    back
    rip
    back
    tcp
    back
  back
  snmp
    community label public
        access read-write
      manager address 0.0.0.0
      back
    back
  back
  telnet
    server
    back
  back
  tftp
      default-volume 5
  back
  ftp
      default-volume 5
  back
back
```

34. Return to root level.

```
ftp# box
box#
```

35. Save the file using a name other than *config* until you can test the configuration.

```
box#  save config startup.cfg
```

36. Test the intial IP interface.

```
box#  ping 192.168.133.114
IP ping: 192.168.133.114 is alive (size = 16 bytes)
```

37. Ensure that the initial IP interface connects to another device on the network.

```
box# ping 192.168.133.97
IP ping: 192.168.133.97 is alive (size = 16 bytes)
```

38. **When you finish configuring the router, exit configuration mode.**

```
box#  exit
bcc>
```

39. **Exit the BCC, which returns you to the Technician Interface prompt.**

```
bcc>  exit
Router1>
```

40. **Enter the logout command to close your console or Telnet session with the router.**

```
Router1> logout
```

Entering Configuration Commands from a File

If you have Manager (read-write) login privileges, you can use the **source** command to read BCC configuration and navigation commands from a designated ASCII source file into the active device configuration.

Caution: The **source** command makes immediate changes to the active device configuration.

The **source** command takes new BCC configuration commands and data from an ASCII file and *merges* those commands with existing configuration data in active device memory. If the file you specify in the **source** command contains configuration commands pertaining to objects already defined on a device, the commands in the file dynamically overwrite the configuration of those objects.

Syntax for the **source** command is as follows:

source *<volume>*:*<filename>*

For information about how to save as an ASCII file the output of the **show config** command or a manually entered set of commands, refer to “Saving Commands on a Workstation” in Chapter 3.

Disabling a Configured Object

In most cases, the BCC automatically enables objects that you add to the device configuration. However, you may need to disable an object to manage or troubleshoot the device. Here is an example of how to disable an object (**rip**) on **ip/1.2.3.4/255.0.0.0**:

1. **Specify the configuration context for the object you want to disable.**

```
box#  ethernet/2/1;ip/1.2.3.4/255.0.0.0;rip
rip/1.2.3.4#
```

2. **Disable RIP.**

```
rip/1.2.3.4#  disable
rip/1.2.3.4#
```

3. **Verify that you disabled RIP.**

```
rip/1.2.3.4#  state
  state disabled
rip/1.2.3.4#
```

You can also disable an object from its parent context, using the following syntax:

disable *<BCC_instance_identifier>*

Example

```
ip/1.2.3.4#  disable rip/1.2.3.4
ip/1.2.3.4/255.0.0.0#
```

Using this method, you remain in the current context after disabling the branch object.

Enabling a Configured Object

If you disable a configured object to manage or troubleshoot the local device, you can use the BCC **enable** command to reenable that object. Here is an example of how to enable an object (**rip**) previously disabled on **ip/1.2.3.4/255.0.0.0** using the **disable** command:

1. **Specify the configuration context for RIP.**

```
box#  ethernet/2/1;ip/1.2.3.4/255.0.0.0;rip
rip/1.2.3.4#
```

2. **Reenable RIP.**

```
rip/1.2.3.4#  enable
rip/1.2.3.4#
```

3. **Verify that you reenabled RIP.**

```
rip/1.2.3.4#  state
  state enabled
rip/1.2.3.4#
```

You can also enable an object from its parent context, using the following syntax:

enable *<BCC_instance_identifier>*

Example

```
ip/1.2.3.4/255.0.0.0#  enable rip/1.2.3.4
ip/1.2.3.4/255.0.0.0#
```

Using this method, you remain in the current context after enabling the branch object.

Deleting a Configured Object

Because of the tree hierarchy, objects on higher branches of the tree depend on the state (and existence) of objects closer to the root of the tree. Deleting an object also deletes anything configured on that object.

Here is an example of how to delete an IP interface from the active device configuration:

1. **Navigate to the object you want to delete:**

```
box#  ethernet/13/1
ethernet/13/1#  ip/192.168.133.114/255.255.255.224
ip/192.168.133.114/255.255.255.224#
```

2. **List all objects configured on the current object:**

```
ip/192.168.133.114/255.255.255.224# lso
arp/192.168.133.114/1   rip/192.168.133.114
```

3. **Delete the object.**

```
ip/192.168.133.114/255.255.255.224#  delete
ethernet/13/1#
```

4. **Verify that you deleted the object.**

```
ethernet/13/1# lso
```
(no objects listed)

Note that ip/192.168.133.114/255.255.255.224 no longer appears in the list of objects configured on ethernet/13/1. With a single **delete** command, the BCC automatically deleted the branch objects (arp/192.168.133.114/1 and rip/192.168.133.114) configured on ip/192.168.133.114/255.255.255.224.

You can also delete an object from its parent context, using the following syntax:

delete *<BCC_instance_identifier>*

Example

```
ip/1.2.3.4/255.0.0.0#  delete rip/1.2.3.4
ip/1.2.3.4/255.0.0.0#
```

Using this method, you remain in the current context after deleting the branch object.

Configuration Command Responses

The BCC completes the configuration task you entered unless a syntax, semantics, or completion error occurs. You can display and verify the configuration by entering the **show config** command.

In certain circumstances, the BCC may be unable to complete a configuration command; this is a completion error. When this occurs, the BCC displays an appropriate error message.

Chapter 5
Configuration Examples

This chapter contains examples that demonstrate how to perform the following configuration tasks:

Task	Page
Identifying Installed Link Modules	5-2
Configuring an Ethernet Interface with IP, ARP, and RIP	5-4
Configuring a HSSI Port with IP	5-7
Configuring a Token Ring Interface with IP and RIP	5-10
Configuring PPP and IP on a Synchronous Interface	5-12
Configuring a FDDI Interface with IP and RIP	5-13
Configuring OSPF and BGP	5-15
Configuring Telnet, FTP, SNMP, and NTP	5-23

Identifying Installed Link Modules

Before you begin configuring a device, you can check the complement of boards installed, as shown in this example:

1. Log on to the AN or BN router as Manager.

```
Bay Networks, Inc. and its Licensors.
Copyright 1992, 1993, 1994, 1995, 1996, 1997. All rights reserved.

Login: Manager
Mounting new volume...
Device label:
Directory: 2:
New Present Working Directory: 2:

      Welcome to the Backbone Technician Interface

Router1>
```

2. Start the BCC by entering bcc at the Technician Interface prompt.

```
Router1> bcc

     Welcome to Bay Command Console!

     * To enter configuration mode, type config

     * To list all system commands, type ?

     * To exit the BCC, type exit

bcc>
```

3. Enter BCC configuration mode by entering config.

```
bcc>  config
box#
```

4. Display the device configuration.

```
box# show config

box type frecn (BCN router with a FRE-2 controller)
  board slot 3
      type sync (Quad synchronous module in slot 3)
  back
  board slot 5
      type hssi (High-Speed Synchronous Interface module in slot 5)
  back
  board slot 7
      type srml (System resource module in slot 7)
  back
  board slot 9
      type dtok (Dual token ring module in slot 9)
  back
  board slot 11
      type wffddi2m (Multimode FDDI module in slot 11)
  back
  board slot 13
      type qenf (Quad Ethernet with filters in slot 13)
    .    .    .    .
    .    .    .    .
    .    .    .    .
```

Refer to the Release Notes for link module descriptions based on any abbreviation used by the BCC.

Configuring an Ethernet Interface with IP, ARP, and RIP

You can add or customize a physical interface and add protocols to that interface as shown in the following example:

1. Log on to the AN or BN router as Manager.

```
Bay Networks, Inc. and its Licensors.
Copyright 1992, 1993, 1994, 1995, 1996, 1997. All rights reserved.

Login: Manager
Mounting new volume...
Device label:
Directory: 2:
New Present Working Directory: 2:

      Welcome to the Backbone Technician Interface

Router1>
```

2. Start the BCC by entering bcc at the Technician Interface prompt.

```
Router1> bcc

     Welcome to Bay Command Console!

     * To enter configuration mode, type config

     * To list all system commands, type ?

     * To exit the BCC, type exit

bcc>
```

3. Enter BCC configuration mode by entering config.

```
bcc>  config
box#
```

4. Add to the device configuration an Ethernet interface on slot 13, connector 1.

```
box# eth 13/1
ethernet/13/1#
```

5. **List the parameters and their assigned values for the current object (ethernet/13/1).**

```
ethernet/13/1# info
  on box
  state enabled
  circuit-name E131
  slot 13
  connector 1
  bofl enabled
  bofl-timeout 5
  hardware-filter disable
  transmit-queue-length 0
  receive-queue-length 0
  bofl-retries 5
  bofl-tmo-divisor 1
  bofl-number 0
```

6. **Change the value of the bofl parameter for ethernet/13/1 from enabled to disabled.**

```
ethernet/13/1# bofl disabled
ethernet/13/1#
```

7. **Verify the new values for bofl and other parameters of ethernet/13/1.**

```
ethernet/13/1# info
  on box
  state enabled
  circuit-name E131
  slot 13
  connector 1
  bofl disabled
  bofl-timeout 5
  hardware-filter disabled
  transmit-queue-length 0
  receive-queue-length 0
  bofl-retries 5
  bofl-tmo-divisor 1
  bofl-number 0
```

8. Add IP (address 192.168.1.1, mask 255.255.255.224) to ethernet/13/1.

```
ethernet/13/1# ip 192.168.1.1/24
ip/192.168.1.1/255.255.255.224#
```

The ".../24" in the mask value represents the number of binary bits reserved for the network portion of the interface address. The BCC also recognizes a mask value you enter in dotted-decimal notation, as follows:

```
ethernet/13/1# ip 192.168.1.1/255.255.255.224
ip/192.168.1.1/255.255.255.224#
```

9. Enable RIP on ip/192.168.1.1/255/255/255/224.

```
ip/192.168.1.1/255.255.255.224# rip
rip/192.168.1.1#
```

Although you typically add RIP at the "ip/..." prompt shown here, you can also enter the **rip** command without navigating to "ip/..." on this branch of the router configuration tree, as follows:

```
arp/192.168.1.1/1# rip
rip/192.168.1.1#
```

The BCC searches backward toward root level until it finds the first level where the object you named can be configured. Finding this level in the tree, the BCC configures the object and leaves you in that context. If the object you name already exists, the BCC merely enters that context.

Note in the second example that the starting context is ARP and the ending context is RIP.

10. Verify values currently assigned to parameters of RIP.

```
rip/192.168.1.1# info
  on ip/192.168.1.1/255.255.255.224
  state enabled
  supply enabled
  listen enabled
  default-supply disabled
  default-listen disabled
   .       .       .
   .       .       .
   .       .       .
```

11. Go back to the previous configuration level.

```
rip/192.168.1.1# back
ip/192.168.1.1/255.255.255.224#
```

Configuring a HSSI Port with IP

To configure a HSSI port with IP, you must also specify a WAN protocol such as Standard, as shown in this example.

1. Log on to the AN or BN router as Manager.

```
Bay Networks, Inc. and its Licensors.
Copyright 1992, 1993, 1994, 1995, 1996, 1997. All rights reserved.

Login: Manager
Mounting new volume...
Device label:
Directory: 2:
New Present Working Directory: 2:

      Welcome to the Backbone Technician Interface

Router1>
```

2. Start the BCC by entering bcc at the Technician Interface prompt.

```
Router1> bcc

     Welcome to Bay Command Console!

     * To enter configuration mode, type config

     * To list all system commands, type ?

     * To exit the BCC, type exit

bcc>
```

3. Enter BCC configuration mode by entering config.

```
bcc>  config
box#
```

4. Configure the single-port HSSI interface on slot 5, connector 1.

```
box# hssi 5/1
hssi/5/1#
```

5. Check values currently assigned to this object.

```
hssi/5/1# info
  on box
  state enabled
  circuit-name H21
  slot 2
  connector 1
  bofl enabled
  bofl-timeout 1
  mtu 4608
  media dsthree
  external-clock-speed 46359642
  crc-size crc32bit
  internal-clock-test disabled
  bofl-number 5
  bofl-length 100
  receive-queue-length 0
  transmit-queue-length 0
  carrier-delay 0
  loopback none
```

6. Determine what you can configure next in this context.

```
hssi/5/1# ?
Sub-Contexts:
  ppp             standard

Parameters in Current Context:
  bofl                 crc-size                name
  bofl-length          external-clock-speed    on
  bofl-number          has                     receive-queue-length
  bofl-timeout         internal-clock-test     slot
  carrier-delay        loopback                state
  circuit-name         media                   transmit-queue-length
  connector            mtu

System Commands:
   .     .     .
   .     .     .
   .     .     .
```

From this context in the active router configuration, you can add PPP or Standard WAN protocol, or you can customize the current value assigned to any parameter of the current object, hssi/5/1.

7. Determine the purpose of these hssi parameters. (Enter the help parameters command.)

```
hssi/5/1# help param

                              hssi Parameters:
_____________________________________________________________________

on
       Identifies the parent(s) of this object.

state
       Indicates the administrative state of this object (enabled or
disabled).

bofl
       Allows the breath-of-life polls to be disabled.

bofl-length
       Specifies the breath of life packet length.

bofl-number
      Specifies the number of breath of life packets per breath of life.

bofl-timeout
       Specifies the number of seconds for the BOFL timer.

    .        .        .        .        .        .
    .        .        .        .        .        .
    .        .        .        .        .        .
```

8. Enter “standard” as the WAN protocol.

```
hssi/5/1# standard
standard/5/1#
```

9. Add an instance of IP (address 192.168.17.1 with the Class C natural mask) to the HSSI port.

```
standard/5/1# ip 192.168.17.1/255.255.255.0
ip/192.168.17.1/255.255.255.0#
```

Configuring a Token Ring Interface with IP and RIP

You can configure IP and RIP on a token ring interface with just three configuration commands, as shown in this example. Note that this example begins at the last configuration level from the previous example (a location other than "box"), to demonstrate a configuration in progress.

1. Configure the dual-port token ring interface on slot 9, connector 1.

```
ip/192.168.17.1/255.255.255.0#  tokenring 9/1
tokenring/9/1#
```

The BCC automatically searches backward toward root to find a level where "tokenring" can be configured. The BCC finds "tokenring" is configurable at root (box) level, completes that command, and leaves you in the context of the newly configured object.

2. Show the configurable values for ring speed.

```
tokenring/9/1# speed ?
Current value: 16Meg
Legal value: {4Meg 16Meg}
Default value: 16Meg
```

3. Set the speed of the token ring interface to 4 Mb/s.

```
tokenring/9/1# speed 4meg
tokenring/9/1#
```

4. Verify the new value for the speed parameter.

```
tokenring/9/1# speed
  speed 4Meg
```

You can always get the current value assigned to any parameter of the current object by entering its name and pressing the Return key.

5. Check to see what you can configure next in this context.

```
tokenring/9/1# ?
Sub-Contexts:
  ip

Parameters in Current Context:
  bofl-number     has             slot
  circuit-name    name            speed
  connector       on              state

System Commands:
   .     .     .
   .     .     .
   .     .     .
```

You can configure parameters of tokenring/9/1 or add an instance of IP on the interface.

6. Add IP (address 192.168.2.1 with the Class C natural mask) to tokenring 9/1.

```
tokenring/9/1# ip 192.168.2.1/24
ip/192.168.2.1/255.255.255.0#
```

7. Enable RIP on ip/192.168.2.1/255.255.255.0.

```
ip/192.168.2.1/255.255.255.0# rip
rip/192.168.2.1#
```

8. Go back to the root-level (box#) prompt.

```
rip/192.168.2.1# box
box#
```

Configuring PPP and IP on a Synchronous Interface

This example configures PPP and IP on a synchronous interface. This example begins in the last configuration context from the previous example. The example excludes multiline and multilink PPP configurations, which you must create using Site Manager.

1. **Add to the device configuration a synchronous interface on slot 3, connector 2.**

```
rip/192.168.2.1# sync 3/2
sync/3/2#
```

2. **Add PPP to sync/3/2.**

```
sync/3/2# ppp
ppp/3/2#
```

3. **Check to see what you can configure next in this context.**

```
ppp/3/2# ?
Sub-Contexts:
  ip      line

Parameters in Current Context:
  has     name    on

System Commands:
   .      .      .
   .      .      .
   .      .      .
```

   You can configure parameters of PPP, or you can add an IP object to ppp/3/2. (The BCC automatically creates a Line object when you add PPP to an interface.)

4. **Add IP (address 192.168.4.1) to ppp/3/2.**

```
ppp/3/2# ip 192.168.4.1 mask 255.255.255.0
ip/192.168.4.1/255.255.255.0#
```

5. **Display the full configuration path from root level to ip/192.168.4.1.**

```
ip/192.168.4.1# pwc
box; sync/3/2; ppp/3/2; ip/192.168.4.1/255.255.255.0;
```

Configuring a FDDI Interface with IP and RIP

This example includes an attempt to change the address assigned to IP on a FDDI interface. Note the BCC responses to this attempt, and note the corrective action.

1. Configure an FDDI interface on slot 11, connector 1.

```
ip/192.168.4.1/255.255.255.0# fddi 11/1
fddi/11/1#
```

The BCC searches recursively back to box context, where it can configure an FDDI interface, does this, and ends at the context of the newly configured object.

2. Add IP (address 192.168.5.1 with the Class C natural mask) to fddi/11/1.

```
fddi/11/1#  ip 192.168.5.1/24
ip/192.168.5.1/255.255.255.0#
```

3. Enable RIP on ip/192.168.5.1/255.255.255.0.

```
ip/192.168.5.1/255.255.255.0# rip
rip/192.168.5.1#
```

4. Display the full path from root level to rip/192.168.5.1.

```
rip/192.168.5.1# pwc
box; fddi/11/1; ip/192.168.5.1/255.255.255.0; rip/192.168.5.1;
```

5. Verify the values currently assigned to the configurable attributes of RIP on ip/192.168.5.1/255.255.255.0.

```
rip/192.168.5.1# info
  on ip/192.168.5.1/255.255.255.0
  state enabled
  supply enabled
  listen enabled
  default-supply disabled
  default-listen disabled
    .    .    .    .    .
    .    .    .    .    .
    .    .    .    .    .
  version rip1
  triggered-updates disabled
  authentication-type none
  authentication {}
  frsvc disabled
```

6. Go back to the previous level in the configuration tree.

```
rip/192.168.5.1# back
ip/192.168.5.1/255.255.255.0#
```

7. Try to change the IP address of the interface.

```
ip/192.168.5.1/255.255.255.0# address 192.168.5.2
"address" attribute may not be modified
ip/192.168.5.1/255.255.255.0#
```

The address cannot be changed because it is a required attribute and part of the instance identifier for IP on an interface.

To change the address assigned to this instance of IP, you must delete the instance and add a new instance of IP with a new address on fddi/11/1, as follows:

```
ip/192.168.5.1/255.255.255.0# delete
fddi/11/1#
```

When you delete IP from an interface, the BCC also deletes RIP from that interface.

8. Add a new instance of IP (with the new address) to fddi/11/1.

```
fddi/11/1# ip 192.168.5.2/24
ip/192.168.5.2/255.255.255.0#
```

9. Exit BCC configuration mode, and then exit to the Technician Interface prompt.

```
ip/192.168.5.2/255.255.255.0#   exit
bcc>  exit
Router1>
```

Configuring OSPF and BGP

This example shows how to

- Add two OSPF areas.
- Add OSPF to interfaces configured on the device.
- Configure BGP.
- Add one BGP peer.
- Add two BGP policy filters.

You could proceed as follows:

1. **Determine what you can configure at box level.**

```
box# ?
Sub-Contexts:
  board           ftp             snmp            tokenring
  console         hssi            sync            virtual
  ethernet        ip              telnet
  fddi            ntp             tftp

Parameters in Current Context:
  build-date      has             name            uptime
  build-version   help-file-name  on
  contact         location        system-name
  description     mib-counters    type

System Commands:
   .     .     .
   .     .     .
   .     .     .
```

2. **Add a virtual IP interface. (You could add a physical interface instead of a virtual interface.)**

```
box# virtual
virtual#
```

   At box level, the BCC allows you to create a virtual interface (an interface not associated with any physical slot on the device).

3. **Add IP (address 192.168.100.1 with the Class C natural mask) to the virtual interface.**

```
virtual#  ip address 192.168.100.1 mask 255.255.255.0
ip/192.168.100.1/255.255.255.0#
```

When IP interfaces configured on physical slots fail, you can still establish communication with an IP interface configured on a virtual interface.

4. **Go back to the root-level (box#) prompt.**

```
ip/192.168.100.1/255.255.255.0# box
```

5. **Return to the context of global IP.**

```
box# ip
ip#
```

6. **Add OSPF to the global IP object.**

```
ip# ospf
ospf#
```

This is the global/box-wide OSPF service object.

7. **Check to see what you can configure next in this context.**

```
ospf# ?
Sub-Contexts:
  accept          announce        area

Parameters in Current Context:
  as-boundary-router      has                     router-id
  as-default-tag          holddown                slot-mask
  ase-metric-support      log-mask                state
  backup-log-mask         name
  backup-lsdb             on

System Commands:
   .     .     .
   .     .     .
   .     .     .
```

You can configure parameters of the global OSPF object, or you can add an OSPF area, accept policy, or announce policy.

8. Check legal values for the as-boundary-router parameter, then specify that this autonomous system (AS) boundary router can receive external RIP routes.

```
ospf# as-boundary-router ?
Current Value: false
Legal Value: {true false}
Default Value: false
```

Instead of reentering the parameter name, just recall the last command using Control+p (or the up arrow), and then change the "?" entry to "true."

```
ospf# as-boundary-router true
```

9. Add an ospf area (number 0.0.0.0) to the global OSPF object.

```
ospf# area 0.0.0.0
area/0.0.0.0#
```

10. Verify the values currently assigned to parameters of OSPF area 0.0.0.0.

```
area/0.0.0.0# info
  on ospf
  state enabled
  area-id 0.0.0.0
  stub false
  authentication-type none
  stub-metric 1
  import-summaries true
```

11. Add another OSPF area (number 0.0.0.1) to the global OSPF object.

```
area/0.0.0.0# area 0.0.0.1
area/0.0.0.1#
```

Instead of entering "area 0.0.0.1," you can also recall the area 0.0.0.0 command (using Control+p or the up arrow key twice), and simply change the last digit of "0.0.0.0" to read "0.0.0.1" and press Return.

12. Verify the values currently assigned to parameters of OSPF area 0.0.0.1.

```
area/0.0.0.1# info
  on ospf
  state enabled
  area-id 0.0.0.1
  stub false
  authentication-type none
  stub-metric 1
  import-summaries true
```

13. Go back to the previous configuration level.

```
area/0.0.0.1# back
ospf#
```

14. Verify the values currently assigned to attributes of the global OSPF object.

```
ospf# info
  on ip
  state enabled
  router-id 192.168.100.1
  slot-mask all-slots
  as-boundary-router true
  holddown 1
  ase-metric-support disabled
  backup-lsdb disabled
  log-mask 287
  backup-log-mask 0
  as-default-tag default
```

The IP address assigned earlier to the virtual interface also serves as the OSPF router ID.

15. Verify the areas configured on the global OSPF object.

```
ospf# lso
area/0.0.0.0 area/0.0.0.1
```

or:

```
ospf# has
  has area/0.0.0.0 area/0.0.0.1
```

16. Add to the device configuration a new Ethernet interface on slot 13, connector 2.

```
ospf# eth 13/2
ethernet/13/2#
```

The BCC searches back toward root to find the context (box), to configure an Ethernet interface.

17. Add IP to the new interface.

```
ethernet/13/2# ip 192.168.8.1/24
ip/192.168.8.1/255.255.255.0#
```

18. Add OSPF area 0.0.0.0 to ip/192.168.8.1/255.255.255.0.

```
ip/192.168.8.1/255.255.255.0# ospf area 0.0.0.0
ospf/192.168.8.1#
```

19. Verify the values currently assigned to attributes of ospf/192.168.8.1.

```
ospf/192.168.8.1# info
  on ip/192.168.8.1/255.255.255.0
  state enabled
  area 0.0.0.0
  authentication {}
  type broadcast
  priority 1
  transit-delay 1
  retransmission-interval 5
  hello-interval 10
  dead-interval 40
  poll-interval 120
  metric 1
  mtu 1
```

20. Add to the device configuration a new Ethernet interface on slot 13, connector 3.

```
ospf/192.168.8.1# eth 13/3
ethernet/13/3#
```

The BCC searches back toward root (box) level to find a level where an Ethernet interface can be configured.

21. Add IP (address 192.168.9.1 with the Class C natural mask) to ethernet/13/3.

```
ethernet/13/3# ip 192.168.9.1/24
ip/192.168.9.1/255.255.255.0#
```

22. Add OSPF area 0.0.0.1 to ip/192.168.9.1/255.255.255.0.

```
ip/192.168.9.1/255.255.255.0# ospf area 0.0.0.1
ospf/192.168.9.1#
```

23. Go back to the previous level in the configuration tree.

```
ospf/192.168.9.1# back
ip/192.168.9.1/255.255.255.0#
```

24. Add ARP to the IP interface, and verify the values that the BCC assigns to parameters of ARP.

```
ip/192.168.9.1/255.255.255.0# arp
arp/192.168.9.1/8# info
  on ip/192.168.9.1/255.255.255.0
  state enabled
```

25. Go back to the root-level (box#) prompt.

```
arp/192.168.9.1/8# box
box#
```

26. Configure a synchronous interface on slot 5, connector 3.

```
box# sync 5/3
sync/5/3#
```

27. Add PPP to sync/5/3.

```
sync/5/3# ppp
ppp/5/3#
```

28. Add IP (address 192.168.10.1 with the Class C natural mask) to ppp/5/3.

```
ppp/5/3# ip 192.168.10.1 mask 255.255.255.0
ip/192.168.10.1/255.255.255.0#
```

29. Return to the root level (box#) prompt.

```
ip/192.168.10.1/255.255.255.0# box
box#
```

30. Return to the context of global IP.

```
box# ip
ip#
```

31. Configure BGP on global IP.

```
ip# bgp
bgp#
```

32. Check to see what you can configure next in this context.

```
bgp# ?
Sub-Contexts:
  accept          announce        debug-control   peer

Parameters in Current Context:
  black-hole-punching     med-comparison          router-id
  has                     multi-hop               slot-mask
  inject-time             name                    state
  intra-as-routing        on                      subnet-aggregation
  local-as                redistribute-protocols
  max-redundant-routes    redundant-connection

System Commands:
   .     .     .
   .     .     .
   .     .     .
```

From this context, you can configure parameters of BGP; you can add BGP accept and announce policies; and you can add a BGP peer or debug-control object to the device configuration.

33. Set the local AS number to 13.

```
bgp# local-as 13
bgp#
```

34. Specify the bgp peer (local address 192.168.10.1, remote address 192.168.10.2) in AS 14.

```
bgp# peer local 192.168.10.1 remote 192.168.10.2 as 14
peer/192.168.10.1/192.168.10.2#
```

35. Verify the values currently assigned to BGP peer/192.168.10.1/192.168.10.2.

```
peer/192.168.10.1/192.168.10.2# info
  on bgp
  state enabled
  local 192.168.10.1
  remote 192.168.10.2
  as 14
  min-version bgp4
  max-version bgp4
  advertise-time 30
  retry 120
    .    .    .    .    .
    .    .    .    .    .
    .    .    .    .    .
```

36. Go back to the previous level in the configuration tree.

```
peer/192.168.10.1/192.168.10.2# back
bgp#
```

37. Add a BGP announce policy called “announce all”.

```
bgp# announce polname "announce_all"
announce/announce_all/bgp#
```

Do not leave any spaces between elements of the policy name; insert either a hyphen or an underscore between elements, as shown in this example.

38. Announce all networks to AS 14.

```
announce/announce_all/bgp# action announce
announce/announce_all/bgp# match
match/bgp/announce/announce_all# outbound-as 14
outbound-as/14/bgp/announce/announce_all#
```

39. Add a BGP accept policy called “accept-192.168.0.0”.

```
outbound-as/14/bgp/announce/announce_all# accept "accept-192.168.0.0"
accept/accept-192.168.0.0/bgp#
```

40. Accept routes from AS 14.

```
accept/accept-192.168.0.0/bgp# action accept
accept/accept-192.168.0.0/bgp# match
match/bgp/accept/accept-192.168.0.0# originating-as 14
originating-as/14/bgp/accept/accept-192.168.0.0#
```

41. Go back to the previous level in the configuration tree. Specify the addresses you want to accept from AS 14.

```
originating-as/14/bgp/accept/accept-192.168.0.0# back
match/bgp/accept/accept-192.168.0.0#  network address 192.168.0.0 mask
255.255.0.0 match range
network/192.168.0.0/255.255.0.0/Range/bgp/accept/accept-192.168.0.0#
```

42. Verify the values currently assigned to this context.

```
network/192.168.0.0/255.255.0.0/Range/bgp/accept/accept-192.168.0.0# info
     on match/bgp/accept/accept-192.168.0.0
     address 192.168.0.0
     mask 255.255.0.0
     match Range
```

Only addresses starting with 192.168.*x.x* will be accepted.

43. Go back to accept/accept-192.168.0.0 in the configuration tree.

```
network/192.168.0.0/255.255.0.0/Range/bgp/accept/accept-192.168.0.0# back
   match/bgp/accept/accept-192.168.0.0# back
   accept/accept-192.168.0.0/bgp#
```

44. Verify the values currently assigned to attributes of the accept policy accept/accept-192.168.0.0.

```
accept/accept-192.168.0.0/bgp# info
  on bgp
  state enabled
  polname accept-192.168.0.0
  action accept
  preference 1
  precedence 0
  bgp4-preference 1
  as-weight-class class1
```

45. Go back to the root-level (box#) prompt.

```
accept/accept-192.168.0.0/bgp# box
box#
```

Configuring Telnet, FTP, SNMP, and NTP

This is an example of how to configure four global/box-wide services on a BN router. These are typical, initial configuration tasks.

1. Add Telnet to the device configuration.

```
box# telnet
telnet#
```

2. Determine what is configurable at this level.

```
telnet# ?
Sub-Contexts:
  client  server

Parameters in Current Context:
  has     name    on      state

System Commands:
   .     .     .
   .     .     .
   .     .     .
```

3. Add Telnet server capability to global Telnet.

```
telnet# server
server#
```

4. Add SNMP to the device configuration.

```
server# snmp
snmp#
```

The BCC searches backward toward root to find the context (box) where it can configure SNMP. The BCC configures SNMP and leaves you in that context.

5. Check to see what you can configure next in this context.

```
snmp# ?
Sub-Contexts:
  community       trap-entity     trap-event

Parameters in Current Context:
  authentication-traps    lock-timeout            state
  has                     name                    type-of-service
  lock                    on
  lock-address            scope-delimiter

System Commands:
   .     .     .
   .     .     .
   .     .     .
```

6. Specify an SNMP community name, "public."

```
snmp# community public
community/public#
```

7. Add manager 0.0.0.0 to community/public.

```
community/public# manager 0.0.0.0
manager/public/0.0.0.0#
```

8. Verify the values currently assigned to parameters of manager/public/0.0.0.0.

```
manager/public/0.0.0.0# info
  on community/public
  address 0.0.0.0
  trap-port 162
  traps generic
```

9. Add TFTP to the device configuration.

```
manager/public/0.0.0.0# tftp
tftp#
```

The BCC searches backward to find a context (box) where it can configure TFTP. The BCC configures TFTP and leaves you in that context.

10. Check to see what you can configure next in this context.

```
tftp# ?
Sub-Contexts:
  <None>

Parameters in Current Context:
  close-timeout           legal-sub-protocols     retry-count
  default-volume          name                    retry-timeout
  has                     on                      state

System Commands:
   .     .     .
   .     .     .
   .     .     .
```

11. Determine the default volume for TFTP.

```
tftp# def
  default-volume 5
```

12. Add FTP to the device configuration.

```
tftp# ftp
ftp#
```

The BCC searches backward to find a context (box) where it can configure FTP. The BCC configures FTP and leaves you in that context.

13. List/define the configurable attributes of global FTP.

```
ftp# help param
                                ftp Parameters:
_____________________________________________________________________

  on
        Identifies the parent(s) of this object.

  state
        Indicates the administrative state of this object (enabled or
        disabled).

  default-volume
        Specifies the default volume where transferred files are
        written/retrieved.

  idle-timeout
       Specifies the timeout in seconds to close the idle FTP session.

  login-retries
        Specifies the number of login retries allowed after a login
        failure.

  max-sessions
        Specifies the maximum number of FTP sessions allowed at one
        time.

  tcp-windowsize
        Specifies the size in bytes of TCP windows used for
        connections.
```

14. Verify the values currently assigned to attributes of global FTP.

```
ftp# info
  on box
  state enabled
  default-volume 5
  login-retries 3
  idle-timeout 900
  max-sessions 3
  tcp-window-size 60000
```

15. Change the default volume to 1.

```
ftp# def 1
ftp#
```

16. Verify the default volume setting.

```
ftp# def
  default-volume 1
```

17. Add global NTP to the device configuration.

```
ftp# ntp
ntp#
```

18. Configure an NTP peer (address 192.168.11.1).

```
ntp# peer address 192.168.11.1
peer/192.168.11.1#
```

19. Go back to the root-level (box#) prompt.

```
peer/192.168.11.1# box
box#
```

20. Exit configuration mode; exit the BCC, and return to the Technician Interface prompt.

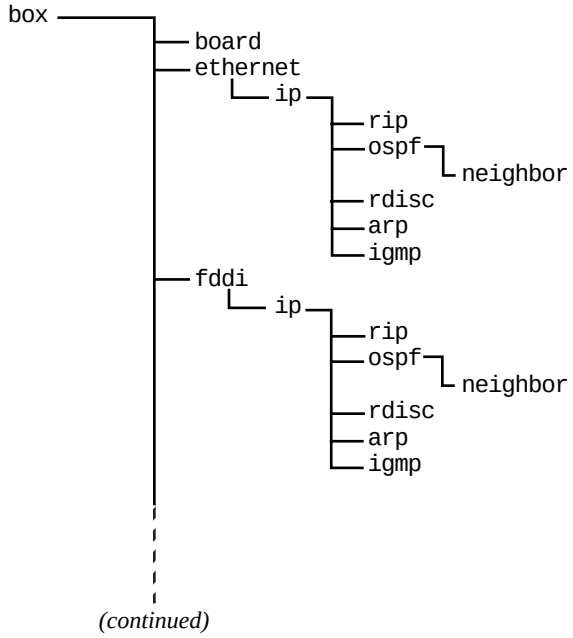
```
box#   exit
bcc>  exit
Router1>
```

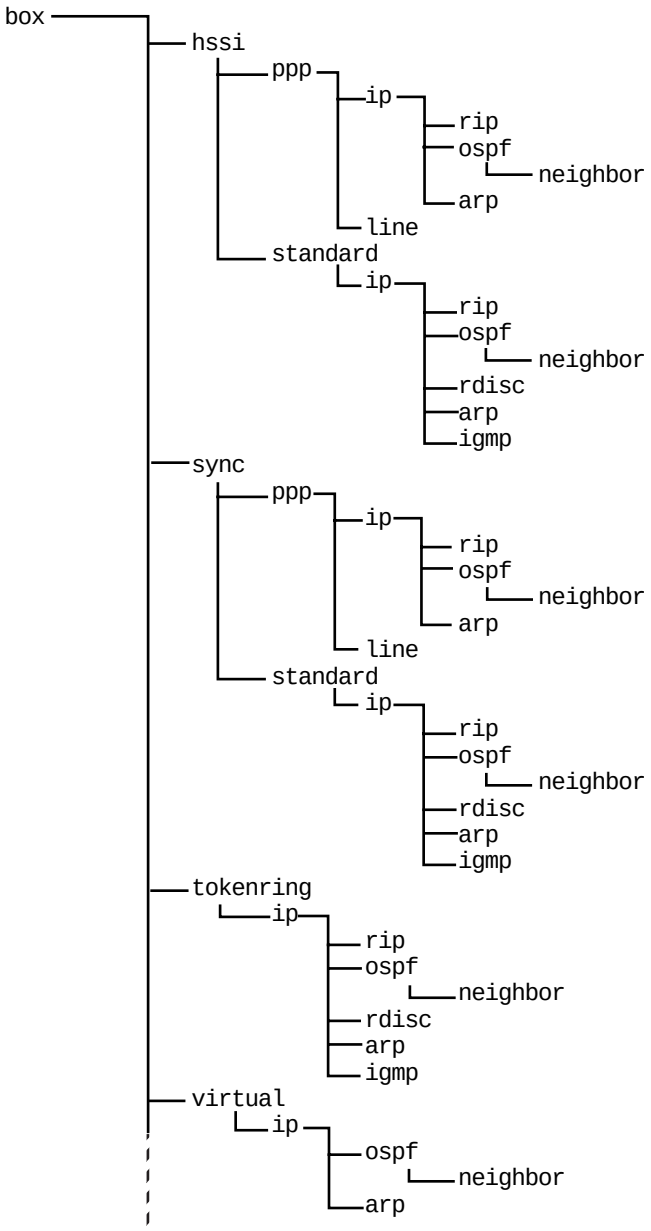
Appendix A
Configuration Tree

You can configure any object shown in the following BCC configuration tree for AN and BN (BLN/BCN) routers. To view all or part of the configuration tree, enter the **help tree** command using the following syntax at any BCC prompt:

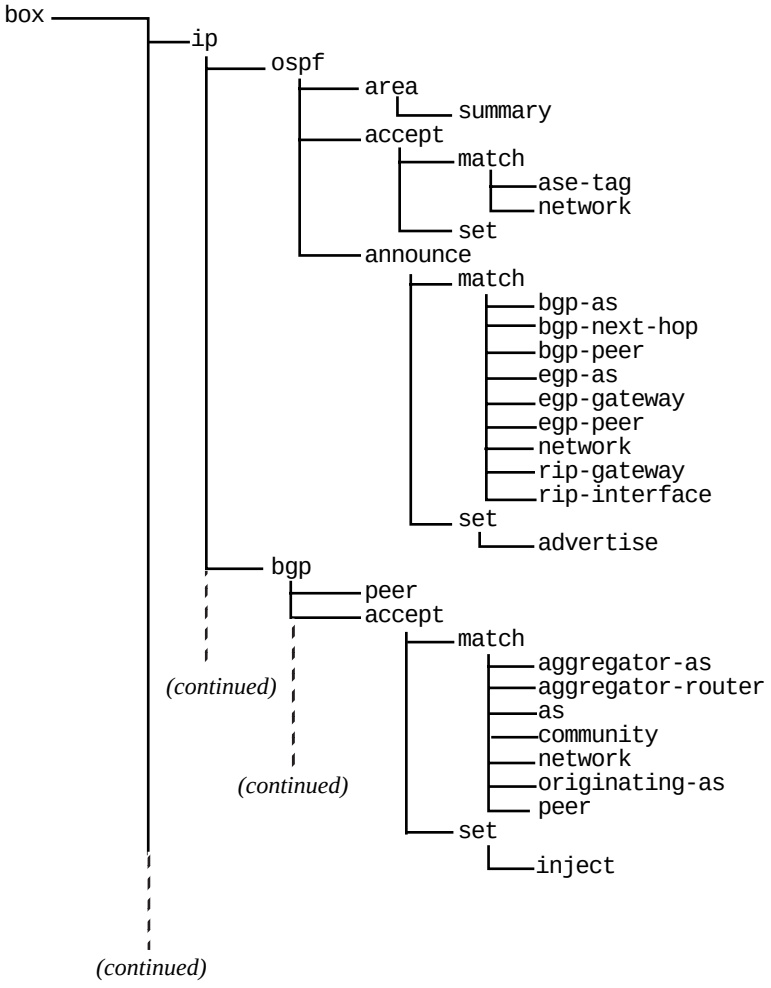
help tree [*<any_box_level_object>*]

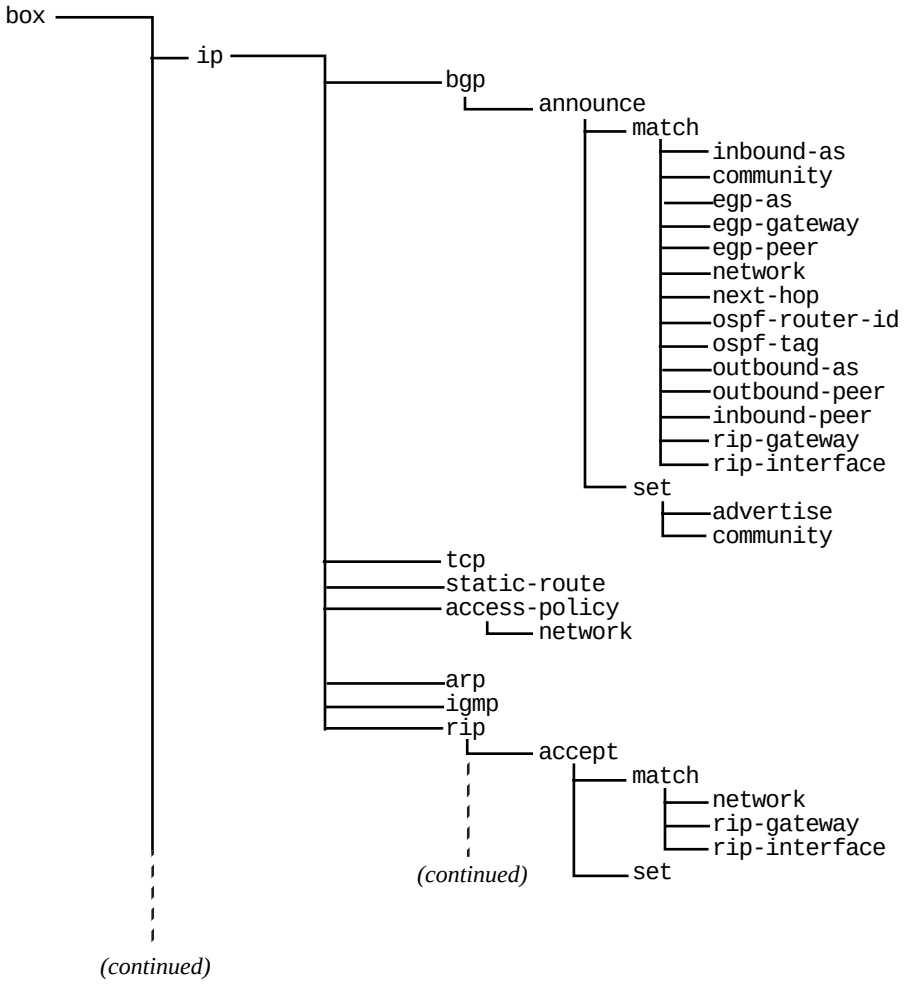
For example: **help tree** (or) **help tree ethernet**

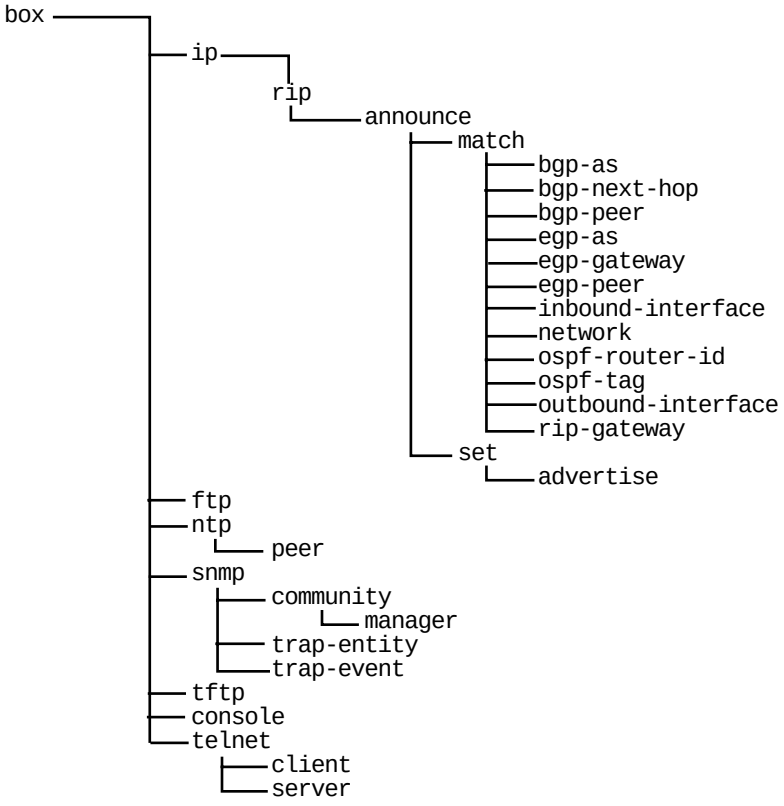



box
hssi
ppp
ip
rip
ospf
neighbor
arp
line
standard
ip
rip
ospf
neighbor
rdisc
arp
igmp
sync
ppp
ip
rip
ospf
neighbor
arp
line
standard
ip
rip
ospf
neighbor
rdisc
arp
igmp
tokenring
ip
rip
ospf
neighbor
rdisc
arp
igmp
virtual
ip
ospf
neighbor
arp

(continued)


box
ip
ospf
area
summary
accept
match
ase-tag
network
set
announce
match
bgp-as
bgp-next-hop
bgp-peer
egp-as
egp-gateway
egp-peer
network
rip-gateway
rip-interface
set
advertise
bgp
peer
accept
match
aggregator-as
aggregator-router
as
community
network
originating-as
peer
set
inject
(continued)
(continued)
(continued)


box
ip
bgp
announce
match
inbound-as
community
egp-as
egp-gateway
egp-peer
network
next-hop
ospf-router-id
ospf-tag
outbound-as
outbound-peer
inbound-peer
rip-gateway
rip-interface
set
advertise
community
tcp
static-route
access-policy
network
arp
igmp
rip
accept
match
network
rip-gateway
rip-interface
set
(continued)
(continued)


box
ip
rip
announce
match
bgp-as
bgp-next-hop
bgp-peer
egp-as
egp-gateway
egp-peer
inbound-interface
network
ospf-router-id
ospf-tag
outbound-interface
rip-gateway
set
advertise
ftp
ntp
peer
snmp
community
manager
trap-entity
trap-event
tftp
console
telnet
client
server

Appendix B
System Commands

AN and BN (BLN/BCN) routers support all system commands shown in Table B-1. Note that some commands perform functions specific to your current interface level (Technician Interface, BCC, or BCC configuration mode) and login privilege level (Manager or User).

Table B-1. AN/BN Router System Commands

Command	Purpose	Login/Access Privileges	
		Manager	User
?	List the names of objects, parameters, and system commands you can enter next.	✔	✔
<command> **?**	Display syntax usage help for *<command>.*		
<object> **?**	List parameters of an object you can access or configure from your current level in the device configuration tree.		
<parameter> **?**	List the current, default, and legal values for this parameter of the current object.		
<object> <parameter> **?**	List the current, default, and legal values for the specified parameter of this object.		
! [*<n>*] (Technician Interface only)	Repeat the last command, or repeat the last command *<n>* times.	✔	✔

(continued)

Table B-1. AN/BN Router System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
back [<*n*>] (BCC configuration mode only)	Move your current working location or level back <*n*> levels closer to the root level of the BCC configuration tree.	✔	✔
bcc	Start the BCC from the Technician Interface prompt.	✔	✔
boot [{<*vol*>:<*image_name*> | - } { <*vol*>:<*config_name*> | - }]	Reboot the system.	✔	
cd [<*vol*>:][<*directory*>]	Set or display the current working volume and directory.	✔	✔
clear <*subcommands*> <*flags*>	Clear specific device information, such as IP and RIP data.	✔	
clearlog [<*slot_ID*>]	Clear all previous events from the system event log.	✔	
commit	Commit (make effective) new values that you assign to MIB attributes.	✔	
compact <*volume*>:	Compact existing files into a contiguous address space on a volume, and compact all unused space into a single contiguous block of free space for new files on the same volume.	✔	

(continued)

Table B-1. AN/BN Router System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
config [**-read-write**|**-read-only**] (BCC only)	Enter BCC configuration mode. If you logged on as Manager, you have read-write access to the device configuration; if you logged on as User, you have read-only access, which allows you to "show" or navigate the existing configuration, but prevents you from changing the configuration. If you enter BCC configuration mode as Manager and want to change your privilege level for the current session from read-write to read-only, enter **config -read-only**. To change Manager privileges back to read-write, enter **config -read-write**.	✔	✔
copy *<vol>:<filename1> <vol>:<filename2>* (Technician Interface only)	Copy the contents of the first file to the second file.	✔	
cp *<vol>:<filename1> <vol>:<filename2>* (BCC only)	Copy the contents of the first file to the second file.	✔	
date [*<mm/dd/yy>*] [*<hh:mm:ss>*] [*<+|-><hh:mm>*]	Display or update the system time and time zone.	✔	✔
delete *<filename> <volume>* (Technician Interface only; see **rm** for BCC equivalent)	Delete the file from the specified volume.	✔	
delete [*<BCC_instance_ID>*] (BCC configuration read-write mode only)	Delete the object described in the current prompt, or delete the object known by the specified *<ID>* and configured previously from the current level.	✔	

(continued)

Table B-1. AN/BN Router System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
diags [*<slot _ID>*]	Perform CPU/memory, backbone, and link diagnostics before downloading the GAME image and rebooting on the specified slot.	✔	
dinfo	Display the status of each file system volume on this device.	✔	✔
dir [*<vol>*:]	Display the contents of the file system volume specified.	✔	✔
disable (BCC configuration read-write mode only)	Disable the object at your current location in the BCC configuration tree.	✔	
enable (BCC configuration read-write mode only)	Enable the object at your current location in the BCC configuration tree.	✔	
format *<volume>*:	Erase the entire contents of the file system volume specified and reinitialize it to a usable state.	✔	
get {*<obj_name>*|*<obj_id>*}.{*<attr_name>*|*<attr_id>*|*}[.{*<inst_id>*|*}] (Technician Interface only)	Retrieve the values of data objects in the MIB.	✔	✔
getcfg (Technician Interface, AN routers only)	Display network boot parameters. (Display the current parameter settings used to determine the source of image and configuration files.)	✔	✔

(continued)

Table B-1. AN/BN Router System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
help [[**commands**] [**-more**] | *<command>* | **syntax** | **tree** [*<box_level_object>*] | **learning-bcc** | **editing**] (BCC only)	The word **help**, entered without arguments, displays an overview of BCC Help-oriented features. That screen describes how to get	✔	✔
help commands	• An alphabetical list of all commands, with syntax and terse descriptions		
help *<command>*	• Detailed Help on a specific command		
help syntax	• Help on how to interpret BCC syntax symbols		
help tree [*<object>*]	• A hierarchical list (the configuration tree) of all objects you can configure on this platform. Entered with the name of a configurable box-level object (like **ip** or **ethernet**), the **help tree** command returns a hierarchical list of objects you can configure in that context. For example, **help tree ip** or **help tree ethernet**.		
help parameters	• A list of parameters of the current object, or of a configurable object described by its path from box level (excluding “box”). For example, **help parameters ip ospf** defines parameters of the global ospf object.		
help [*<task>*]	• Help on how to perform a specific task. The tasks (and their corresponding **help** commands) are **help learning-bcc** **help editing**		

(continued)

Table B-1. AN/BN Router System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
help [*<command>* | **-all**] (Technician Interface only)	The **help** command, entered without arguments at the Technician Interface prompt, displays an alphabetical list of all commands with syntax only. The list excludes commands available only in BCC mode. The combination **help** *<command>* displays detailed Help on a specific command, excluding any commands available only in BCC mode. The combination **help -all** displays, in scrollable format, detailed Help on all system commands, excluding those available only in BCC mode.	✔	✔
history [*<n>*]	Display the command history list or recall command number *<n>* from the history list.	✔	✔
info (BCC configuration mode only)	List values currently assigned to all configurable parameters of this object.	✔	✔
ip *<subcommand> <flags>* (Technician Interface only)	Display data from IP; show a different view for each subcommand or flag you enter.	✔	✔
ip6 *<subcommand> <flags>* (Technician Interface only)	Display data from IPv6; show a different view for each subcommand or flag you enter.	✔	✔
list [[*<instances>* [*<obj_name>*]]] (Technician Interface only)	List objects in the MIB.	✔	✔
loadmap [*<slot_list>* | **all**] [*<filepath>*]	Display the load address and size of each dynamically loadable application (for example, a protocol).	✔	

(continued)

Table B-1. AN/BN Router System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
log [*<vol>*:*<logfile>*] [**-d***<date>*] [**-t***<time>*] [**-e"***<entity>***"**] [**-f***<severity>*] [**-s***<slot_ID>*] [**-p**[*<rate>*] [**-c***<code #>*]	Display the current system event log. Follow the optional **-e** flag immediately with the entity name in quotes and uppercase characters (no intervening spaces). Specify *<severity>* using letters and no intervening spaces: f = fault w = warning i = info t = trace d = debug **Examples:** -fwid -ffitd -fwi -fwitd Use the optional **-p** flag to set an interval for polling the log and displaying the result.	✔	✔
log [**-x**|**-i**] [**-e"***<entity>"*] [**-f***<severity>*] [**-s***<slot_ID>*]	Excludes (**-x**) or includes (**-i**) event logging indicated by the command options.	✔	✔
log -z [**-s***<slot_ID>*]	Displays current filter setting.	✔	✔
logout	Exit the current login session.	✔	✔
lso [**-l**] (BCC configuration mode only)	List next-level (branch) objects configured on the current object. The optional **-l** flag causes the BCC to list object IDs in wrap-around screen format.	✔	✔
mget {*<obj_name>* | *<obj_id>*} . {*<attr_name>* | *<attr_id>* | * } [. {*<inst_id>*|*}] (BCC only)	Retrieve the values of data objects in the MIB.	✔	✔
mlist [[*<instances>* [*<obj_name>*]]] (BCC only)	List objects in the MIB.	✔	✔
more [**on** | **off**] [*<#_lines_per_screen>*]	Set or display the status of the **more** utility.	✔	✔

Table B-1. AN/BN Router System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
			(continued)
mset {*<obj_name>*|*<obj_id>*}**.**{*<attr_name>*| *<attr_id>*}.{*<inst_id>*} {*<value>*} (BCC only)	Modify (set) the values of data objects in the device MIB.	✔	
partition {create|**delete}** [*<vol>***:**]	Create or delete a partition on existing file system media.	✔	✔
password Manager	Change the password of the Manager account.	✔	
password User	Change the password of the User account.	✔	✔
ping -*<protocol> <address>* [**-t***<timeout>*]	Initiate an ECHO request/reply handshake.	✔	✔
pktdump *<line_number>* [**-s***<start>*] [**-c***<count>*]	Display packets that have been captured by an interface configured for Packet Capture.	✔	
prom [**-v**|**-w**] *<vol>*:*<ROM_Update_File><slot_ID>* [*<slot_ID>*...]	Update or verify the software located on a flash PROM device.	✔	
pwc (BCC configuration mode only)	Display the path to your current working location or level in the tree, starting from root level.	✔	✔
readexe *<vol>*:*<filename>*	Validate the checksums of an executable image and print out all the file header information.	✔	✔
record {**open**|**close**} [**-fileonly**] [**-pause**] *<vol>*:*<filename>* (Technician Interface only)	Record to a file all messages written to the terminal. You can open, pause, and close a recording session.	✔	
reset [*<slot_ID>*]	Reboot the GAME image on the specified slot. If the slot ID argument is absent, reboot the entire device.	✔	
restart [*<slot_ID>*]	Restart the GAME image on the specified slot. If the slot ID argument is absent, the GAME image restarts on all slots.	✔	

Table B-1. AN/BN Router System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**

(continued)

rm *<vol>***:***<filename>* (BCC only)	Remove (delete) the file from the specified volume.	✔	
save {**config**|**aliases**} *<vol>***:***<filename>*	Store the current configuration, alias list, or system event.	✔	
securelogin	Turn SecurID access to the device on and off via Telnet.	✔	
set { *<obj_name>* | *<obj_id>* } .{*<attr_name>*| *<attr_id>* } . {*<inst_id>* } {*<value>*} ... (Technician Interface only)	Modifies data objects in the MIB.	✔	
show *<entity> <option>* **show config** [*<BCC_instance_ID>*] (BCC only)	Display statistical information associated with the specified entity and option. Show the total device configuration, or show the configuration of the configured object known by this BCC instance ID.	✔	✔
source *<vol>***:***<filename>* (BCC configuration mode only)	Read BCC configuration commands from a text file, dynamically changing the active device configuration.	✔	
source {aliases | env | perm } *<vol>*:*<filename>* (Technician Interface only)	Read a list of aliases, environment variables, or dynamic permissions from a file.		
stamp	Display the device image version name and timestamp.	✔	✔
system (Technician Interface only)	Start a new Technician Interface session that allows you to run system manager privileged commands.	✔	✔
telnet [**-d**] [**-e** *<escape_char>*] [*<host_ip>* [*<port>*]] (Technician Interface and BCC top level only; not available in BCC configuration mode)	Communicate with other hosts supporting the Telnet protocol.	✔	✔

(continued)

Table B-1. AN/BN Router System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
tftp {**get**|**put**} *<address> <vol>:<file_name>* [*<vol>:<file_name>*] (Technician Interface and bcc> top level only)	Send files to, or retrieve files from, other hosts supporting TFTP.	✔	
type [**-x**] *<vol>:<filename>*	Display the contents of the designated file in ASCII or HEX (**-x**) format.	✔	✔
xmodem {**rb**|**sb}** [**y**|**w**|**p**|**n**] *<filename>* **...**	Transfer files to or from this device over a dial (out-of-band) connection.	✔	

For more detailed information about a specific command:

- Enter **help** <*command*> at any BCC prompt (for example, **help save**).
- Refer to the guide *Using Technician Interface Software*.

Appendix C
TCL Support

The BCC supports the following subset of Tool Command Language (TCL) scripting commands on the router platform:

- append
- break
- case
- catch
- close
- concat
- continue
- eof
- error
- eval
- exit
- expr
- flush
- for
- foreach
- gets
- global
- if
- incr
- interp
- join
- lappend
- lindex
- linsert
- list
- llength
- lrange
- lreplace
- lsearch
- lsort
- open
- proc
- puts
- rename
- return
- set
- source
- split
- subst
- switch
- unset
- uplevel
- upvar
- while

For more information about definitions, syntax, and applications for these TCL commands, refer to the following book:

Ousterhout, J. *Tcl and the Tk Toolkit*. Reading, Mass.: Addison-Wesley, ISBN 0-201-63337-X.

Index

A

abbreviating commands and parameters, 1-7

acronyms, xv

ASCII files, 3-4, 3-14

B

Bay Networks Press, xvi

BCC
- accessing, 1-1, 2-1
- configurable objects, 1-2
- configuration cycle, 4-4
- console types, 2-1
- defined, 1-1
- error messages, 4-20
- Help, 2-12, B-5
- navigating in configuration mode, 2-7
- platform requirements, 1-2
- terminology and concepts, 1-3

board, defined, 1-6

box, defined, 1-6

box-wide objects, 1-6

C

class, defined, 1-3

command line
- adding comments to, 3-6
- continuing, 3-4

command operators
- defined, 3-11
- delete, 3-12, 4-19
- disable, 3-12, 4-17
- enable, 3-12, 4-18
- info, 3-12

commands
- ?, 2-12
- back, 2-7
- config, 1-2
- configuration, 3-7
- editing, 2-13, 3-13
- entering, 3-1
- Help, 2-12
- info, 3-12
- list of all system, 3-6, B-1
- lso, 2-26, 2-28
- pwc, 2-6
- show config, 2-26, 2-28, 2-29
- source, 3-3

configuration
- abbreviated syntax, 3-9
- commands, 3-7
- context-specific, 2-28
- data, 2-26
- default syntax, 3-8
- examples, 5-1
- full syntax, 3-7
- hierarchy, 1-4
- method of, 4-2
- new, 4-1
- total, 2-26

connector, 1-6

context
- changing, 2-7
- current, 2-6
- defined, 1-5, 2-5
- displaying, 2-6

context-sensitive prompt, 2-6

continuing a command line, 3-4

conventions, xiv
creating a new configuration, 4-1
customer support
  programs, xvi
  Technical Solutions Centers, xvii

D

delete command (operator), 3-12, 4-19
derived parameters, 1-3
disable command (operator), 3-12, 4-17

E

editing commands, 3-13
enable command (operator), 3-12, 4-18
entering the BCC, 2-1
error messages, 4-20
examples
  identifying link modules, 5-2
  IP and RIP on FDDI, 5-13
  IP and RIP on Token Ring, 5-10
  IP on HSSI, 5-7
  IP, ARP, and RIP on Ethernet, 5-4
  OSPF and BGP, 5-15
  PPP, IP, and Adjacent Host on Sync, 5-12
  Telnet, FTP, SNMP, and NTP, 5-23
exiting the BCC, 2-1

G

global objects, defined, 1-6

H

has parameter
  described, 1-5, 2-24
  example, 2-28
Help
  commands, 2-12, B-5
  overview, 2-12
history buffer, 3-2

I

info command, 3-12
instance identifier, 1-3, 2-6, 2-10, 3-10
interface, defined, 1-7

L

line, defined, 1-6
list
  command operators, B-1
  command symbols or conventions, B-5
  configurable objects, B-5
  configured objects, B-7
  editing commands, 3-13
  parameter definitions, 2-13
  statistical information, B-9
  system commands, B-1
location in configuration hierarchy, 2-6
login privileges and procedures, 2-2
lso command, 2-26, 2-28

M

Manager/User privilege levels, 2-2
messages, error, 4-20
methods, 3-1, 3-12
modifying a configuration with file-based commands, 4-16

N

navigating
  back one or more levels, 2-8
  forward one level, 2-9
  using configuration commands, 2-8
  using the back command, 2-7

O

objects
- defined, 1-3
- deleting, 4-19
- disabling, 4-17
- enabing, 4-18
- leaf, 1-5

on parameter, 2-24

online Help, BCC, 2-12, B-5

optional attributes, 1-3

P

parameters
- abbreviating, 1-7
- defined, 1-3
- derived values, 1-3
- getting Help, 2-13
- has, 1-5, 2-24
- help definitions for, B-5
- on, 2-24
- required and optional, 3-10
- specifying multiple, 3-11
- specifying values for, 3-10

path, specifying, 2-10

port, defined, 1-6

prompt, context-sensitive, 2-6

protocol, defined, 1-7

publications, ordering, xvi

pwc command, 2-6

R

reading commands from a file, 3-3

recalling commands, 3-2

required attributes, defined, 1-3, 3-8

S

saving configuration commands, 3-14

show config
- command, 2-26
- example, 2-28

slot, defined, 1-6

source command
- defined, 3-3, 4-16
- syntax, 3-4

specifying a path, 2-10

syntax
- getting Help on, 2-12
- types, 3-1

T

TCL support, C-1

Technical Solutions Centers, xvii

Technician Interface, 1-1, 1-2

U

User privileges, 2-2